Selected Topics on Distributed Video Coding

THÈSE N^{O} 4266 (2009)

PRÉSENTÉE LE 9 JANVIER 2009

À LA FACULTE SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
MULTI MEDIA SIGNAL PROCESSING GROUP (MMSPG)
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Mourad OUARET

acceptée sur proposition du jury:

Prof. S. Süsstrunk, présidente du jury
Prof. T. Ebrahimi, directeur de thèse
Prof. R. Leonardi, rapporteur
Prof. M. A. Shokrollahi, rapporteur
Prof. S. Tubaro, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2008

Contents

1 Introduction **1**

2 Distributed Video Coding (DVC) **4**

2.1 Introduction 4
2.2 Theoretical DVC 5
2.3 DVC Implementations 7
2.3.1 PRISM Codec 7
2.3.2 The Stanford Codec 7
2.3.3 DVC Tools 8
2.4 Practical DVC 9
2.4.1 DVC Encoder 10
2.4.2 DVC Decoder 17
2.5 Performance Evaluation of DVC 23
2.6 Applications for DVC 30
2.6.1 DVC Functional Benefits 30
2.6.2 DVC Application Scenarios 30
2.7 Conclusion 33

3 Intra Frame Encoding **35**

3.1 Introduction 35
3.2 JPEG 37
3.3 JPEG2000 37
3.4 Advanced Video Coding (AVC)/H.264 Intra 40
3.5 JPEG XR 42
3.6 Simulation Results 44
3.6.1 Test Material and Evaluation Methodology 44
3.6.2 RD Performance 48
3.7 Conclusion 57

4 Monoview DVC **58**

4.1 Introduction 58
4.2 Related Work 59
4.2.1 Improved Side Information 59
4.2.2 Error Concealment (EC) 61
4.3 Proposed Improved Side Information 62
4.3.1 Matching Criterion 63
4.3.2 Suspicious Vector Detection 64
4.3.3 Motion Vector Refinement and Smoothing 64

4.3.4 Optimal Motion Compensation Mode Selection 65
4.4 Hybrid Error Concealment (EC) 66
4.4.1 Spatial Error Concealment Based on an Edge Directed Filter 66
4.4.2 Enhanced Temporal Error Concealment 68
4.5 Simulation Results 68
4.5.1 Improved SI Performance 68
4.5.2 EC Performance 73
4.6 Conclusion 81

5 Multiview Distributed Video Coding 82
5.1 Introduction 82
5.2 Multiview DVC (MDVC) 83
5.3 Inter-Camera Prediction 86
5.3.1 Disparity Compensation View Prediction (DCVP) 86
5.3.2 Homography 86
5.3.3 View Synthesis Prediction (VSP) 88
5.3.4 View Morphing (VM) 88
5.3.5 MultiView Motion Estimation (MVME) 91
5.4 Iterative Multiview Side Information (IMSI) 92
5.4.1 DVC Reconstruction 93
5.4.2 IMSI for Enhanced Reconstruction 93
5.5 Simulation Results 94
5.5.1 Test Material and Evaluation Methodology 94
5.5.2 Side Information Quality 95
5.5.3 Side Information complexity 99
5.5.4 RD Performance 101
5.5.5 Fusion-based Side Information 104
5.5.6 Comparison with AVC/H.264 107
5.6 Conclusion 109

6 Scalable Distributed Video Coding 111
6.1 Introduction 111
6.2 Scalable Distributed Video Coding (SDVC) 112
6.2.1 Scalable DVC for Image Coding 113
6.2.2 Scalable DVC for Video Coding 115
6.3 JPEG2000 116
6.4 Simulation Results 118
6.4.1 Test Material and Conditions 118
6.4.2 Error-free conditions 120
6.4.3 Error-prone conditions 130
6.5 Conclusion 134

7 DVC Privacy Enabling by Transform Domain Scrambling 135
7.1 Introduction 135
7.2 Scrambling for Distributed Video Coding 136
7.2.1 AVC/H.264 Intra Scrambling 137
7.2.2 Wyner Ziv Frames Scrambling 137
7.2.3 The DCT Coefficient Scrambler 138
7.3 Security Issues 138

7.4 RD Performance 139
7.5 Conclusion 140

8 DVC Demonstrator 141
8.1 Introduction 141
8.2 Demonstrator Design 142
8.2.1 Client Demonstrator 143
8.2.2 Server Demonstrator 143
8.2.3 Network Transmission 144
8.3 Video Decoding Optimization 145
8.3.1 Decoder Profiling 145
8.3.2 Optimization Solutions 146
8.3.3 Implementation and Evaluation 147
8.3.4 Optimization results 148
8.4 Practical Implementation 148
8.4.1 Software Options 149
8.4.2 Description of the Graphical User Interface 150
8.4.3 Decoding performance 154
8.5 Conclusion 156

9 Conclusion 157

Résumé

Le codage vidéo distribué est le nouveau paradigme pour la compression basé sur les théorèmes établis par Slepian et Wolf, et Wyner et Ziv. Alors que les méthodes de codage conventionnel ont une rigide répartition de complexité comme les taches les plus complexes sont faites l'encodeur, le codage distribué rends flexible la répartition de la complexité entre l'encodeur et le décodeur. Le cas le pus intéressant est celui de simples encodeurs et de décodeurs plutt complexes qui est l'opposé du codage conventionnel. Ce dernier est intéressant pour des applications ou le coût du décodeur est plus critique que celui de l'encodeur. Par contre, le codage distribué ouvre la porte une nouvelle gamme d'applications où les encodeurs sont simples et la complexité du décodeur n'est pas si critique. Ceci est très intéressant cause de l'utilisation récurrente de petits appareils multimédia mobiles fonctionnant sur batterie.

Le codage distribué fonctionne comme un système avec complexité inversée en le comparant avec le codage conventionnel. De plus, il donne la possibilité de construire des systèmes avec encodeurs et décodeurs simples. Ceci est possible grce au transcodage, qui transforme le flux généré par le codage distribué en un flux conventionnel. Ce dernier, étant conventionnel, peut donc être décoder de manière simple.

Les systèmes multi-camera sont intéressants pour différentes applications comme la video surveillance. Les différentes vues des cameras peuvent être utilisées pour améliorer la performance des algorithmes de détection d'événements ou d'intrusions. Alors que les méthodes de compression conventionnel exploitent la corrélation entre les différentes cameras l'encodeur, le codage distribué permet de compresser le flux de chaque camera indépendamment des autres cameras. Dans ce cas, une communication entre les différentes cameras n'est pas requise, ce qui est un avantage.

Le codage distribué, étant basé sur une approche statistique, est plus performant que le codage conventionnel dans le cas de transmission travers des environnements erronés. De plus, le codage distribué permet de construire des codecs scalables où la couche de base est indépendante des couches supérieures.

Cette thèse traite les sujets suivants:

Initialement, les fondations théoriques du codage distribué et son implémentation sont présentés. De plus, ses applications sont identifiées.

Le codage distribué utilise le codage conventionnel pour compresser une partie de la video. Pour cela, différents algorithmes de compression conventionnels sont comparés en termes d'efficacité. Les différents paramètres sont choisis de sorte que chaque algorithme produit la meilleure performance possible.

Des outils pour améliorer la prédiction du coté du décodeur sont proposé pour améliorer la performance du codage distribué. Le gain est plus important pour les vidéos avec beaucoup de mouvement. De plus, un algorithme pour dissimuler les erreurs pour le codage distribué en cas de perte de paquets de transmission est introduit. Plus spécifiquement, une technique de dissimulation spatiale est utilisée pour améliorer l'algorithme de dissimulation temporel. La combinaison des deux techniques de dissimulation est supérieure chaque technique appliquée toute seule. Le codage distribué dissimulé est supérieur au codage conventionnel dissimulé dans le cas d'erreurs de transmission.

Les différentes techniques de prédiction pour le codage distribué dans le cas d'un scénario multi-caméra sont comparées en termes de qualité de prédiction, complexité et taux de compression. De plus, une technique de prédiction itérative est introduite pour améliorer la performance pour le contenu vidéo avec beaucoup de mouvement. Des algorithmes de fusion entre différentes prédictions sont proposés aussi pour une meilleure performance.

Le codage distribué est aussi utilisé pour construire des codecs scalables avec les différentes couches étant indépendantes. Ceci donne beaucoup de flexibilité pour distribuer les différentes couches sur le réseau. Les algorithmes proposés traitent la scalabilité temporel et spatiale. De plus, les algorithmes basés sur le codage distribué montrent une résistance aux erreurs de transmission.

La protection d'identité pour des applications comme la vidéo surveillance, où le codage distribué peut être utilisé, est important. Pour cela, une technique de brouillage des régions d'intérêt est proposée en altérant les coefficients transformés dans ces régions.

Finalement, un démonstrateur implémenté durant ce travail de thèse est décrit ou les différents requis et les limitations observées sont présentées. a montre qu'il est possible d'implémenter un système complet de codage distribué en se basant sur des hypothèses réalistes.

Même si le codage distribué est inferieur au codage conventionnel pour le moment, les forces de celui-ci résident dans sa bonne résistance aux erreurs de transmission.

Mots-Clés: Compression Distribué, Robustesse, Scalabilité.

Abstract

Distributed Video Coding (DVC) is a new paradigm for video compression based on the information theoretical results of Slepian and Wolf (SW), and Wyner and Ziv (WZ). While conventional coding has a rigid complexity allocation as most of the complex tasks are performed at the encoder side, DVC enables a flexible complexity allocation between the encoder and the decoder. The most novel and interesting case is low complexity encoding and complex decoding, which is the opposite of conventional coding. While the latter is suitable for applications where the cost of the decoder is more critical than the encoder's one, DVC opens the door for a new range of applications where low complexity encoding is required and the decoder's complexity is not critical. This is interesting with the deployment of small and battery-powered multimedia mobile devices all around in our daily life.

Further, since DVC operates as a reversed-complexity scheme when compared to conventional coding, DVC also enables the interesting scenario of low complexity encoding and decoding between two ends by transcoding between DVC and conventional coding. More specifically, low complexity encoding is possible by DVC at one end. Then, the resulting stream is decoded and conventionally re-encoded to enable low complexity decoding at the other end.

Multiview video is attractive for a wide range of applications such as free viewpoint television, which is a system that allows viewing the scene from a viewpoint chosen by the viewer. Moreover, multiview can be beneficial for monitoring purposes in video surveillance. The increased use of multiview video systems is mainly due to the improvements in video technology and the reduced cost of cameras. While a multiview conventional codec will try to exploit the correlation among the different cameras at the encoder side, DVC allows for separate encoding of correlated video sources. Therefore, DVC requires no communication between the cameras in a multiview scenario. This is an advantage since communication is time consuming (i.e more delay) and requires complex networking.

Another appealing feature of DVC is the fact that it is based on a statistical framework. Moreover, DVC behaves as a natural joint source-channel coding solution. This results in an improved error resilience performance when compared to conventional coding. Further, DVC-based scalable codecs do not require a deterministic knowledge of the lower layers. In other words, the enhancement layers are completely independent from the base layer codec. This is called the codec-independent scalability feature, which offers a high flexibility in the way the various layers are distributed in a network.

This thesis addresses the following topics:

First, the theoretical foundations of DVC as well as the practical DVC scheme used in this research are presented. The potential applications for DVC are also outlined.

DVC-based schemes use conventional coding to compress parts of the data, while the rest is compressed in a distributed fashion. Thus, different conventional codecs are studied in this research as they are compared in terms of compression efficiency for a rich set of sequences. This includes fine tuning the compression parameters such that the best performance is achieved for each codec.

Further, DVC tools for improved Side Information (SI) and Error Concealment

(EC) are introduced for monoview DVC using a partially decoded frame. The improved SI results in a significant gain in reconstruction quality for video with high activity and motion. This is done by re-estimating the erroneous motion vectors using the partially decoded frame to improve the SI quality. The latter is then used to enhance the reconstruction of the finally decoded frame. Further, the introduced spatio-temporal EC improves the quality of decoded video in the case of erroneously received packets, outperforming both spatial and temporal EC. Moreover, it also outperforms error-concealed conventional coding in different modes.

Then, multiview DVC is studied in terms of SI generation, which differentiates it from the monoview case. More specifically, different multiview prediction techniques for SI generation are described and compared in terms of prediction quality, complexity and compression efficiency. Further, a technique for iterative multiview SI is introduced, where the final SI is used in an enhanced reconstruction process. The iterative SI outperforms the other SI generation techniques, especially for high motion video content. Finally, fusion techniques of temporal and inter-view side informations are introduced as well, which improves the performance of multiview DVC over monoview coding.

DVC is also used to enable scalability for image and video coding. Since DVC is based on a statistical framework, the base and enhancement layers are completely independent, which is an interesting property called codec-independent scalability. Moreover, the introduced DVC scalable schemes show a good robustness to errors as the quality of decoded video steadily decreases with error rate increase. On the other hand, conventional coding exhibits a cliff effect as the performance drops dramatically after a certain error rate value.

Further, the issue of privacy protection is addressed for DVC by transform domain scrambling, which is used to alter regions of interest in video such that the scene is still understood and privacy is preserved as well. The proposed scrambling techniques are shown to provide a good level of security without impairing the performance of the DVC scheme when compared to the one without scrambling. This is particularly attractive for video surveillance scenarios, which is one of the most promising applications for DVC.

Finally, a practical DVC demonstrator built during this research is described, where the main requirements as well as the observed limitations are presented. Furthermore, it is defined in a setup being as close as possible to a complete real application scenario. This shows that it is actually possible to implement a complete end-to-end practical DVC system relying only on realistic assumptions.

Even though DVC is inferior in terms of compression efficiency to the state of the art conventional coding for the moment, strengths of DVC reside in its good error resilience properties and the codec-independent scalability feature. Therefore, DVC offers promising possibilities for video compression with transmission over error-prone environments requirement as it significantly outperforms conventional coding in this case.

Key-Words: Distributed Video Coding, Intra Coding, Multiview, Error Concealment, Iterative Side Information, Privacy, Codec-Independent Scalability.

Chapter 1

Introduction

Nowadays, image and video applications are widely used by people on a regular basis. This encouraged the deployment of multimedia products such as digital TV, mobile phones, cameras and Internet. Moreover, the need for video surveillance cameras has rapidly increased in the past years due to terrorist threats and increasing criminality rate. Most of the time, these applications require storage and/or transmission over a wired or wireless environment. This makes compression important, where video is represented in a way that requires less storage capability without significantly degrading the visual quality. At the same time, transmission over error-prone environments has become very common with the increase of wireless devices such as mobile phone devices and wireless cameras. Thus, compression algorithms with good error resilience properties are needed. Moreover, such devices are very likely of being small, limiting therefore the available power and computational resources. For practical reasons, such terminals cannot afford to run complex routines. Therefore, low complexity processing is interesting as it results in low power consumption and simple implementation.
Today's digital video coding schemes are represented by the ITU-T and MPEG [1] standards, which rely on a combination of block-based transform and interframe prediction to exploit spatial and temporal correlations within encoded video. This results in high complexity encoders because of the motion estimation process run at the encoder side. On the other hand, the resulting decoders are simple and around five to ten times less complex than the corresponding encoders [2]. This type of architecture is well-suited for down link model applications such as broadcasting and video-on-demand, where the cost of the decoder is critical. Moreover, these schemes are based on a deterministic approach resulting in a high sensitivity with respect to errors. Indeed, the presence of a prediction loop favors error propagation causing drift effect.

Distributed Source Coding (DSC) has emerged as a technology that enables a flexible complexity allocation between the encoder and the decoder [2]. DSC refers to independently coding correlated sources. Then, the compressed streams are transmitted to a decoder, which exploits the source statistics to perform joint decoding. Initially, Slepian and Wolf established the bounds for the achievable rates in [3] for the lossless case. Then, Wyner and Ziv defined the same bounds for the lossy case in [4].

Applying DSC to video gave birth to a new paradigm for video compression called Distribute Video Coding [5, 6] targeting both, low complexity encoding and error resilience. Most of DVC tools developed in the literature are based on the DVC schemes introduced in [5,6]. PRISM is DVC's practical implementation by the Berkeley's group [5], which is a block-based approach using coset coding [7,8]. On the other hand, the Stanford approach [9] is applied on a frame level and uses turbo codes [10] for distributed coding. Furthermore, it requires a feedback mechanism while PRISM estimates the rate at the encoder side.

Initially, the theoretic foundations of DVC as well as its practical implementation are described in chapter 2. After a detailed description of the implementation of the practical DVC scheme used in this research, its evaluation in error-free and error-prone conditions is also presented. Finally, some promising applications for which DVC is suitable and attractive are described at the end of the chapter. These are mainly applications with low complexity and power requirements in addition to transmission over error-prone environments.

DVC encodes parts of the input video in a traditional way using conventional Intra codecs such as JPEG [11], JPEG2000 [12], AVC/H.264 Intra [13] or JPEG XR [14]. Chapter 3 presents a Rate Distortion (RD) performance evaluation of these codecs using the Peak Signal to Noise Ratio (PSNR) distortion metric. A rich set of test sequences with different spatial resolutions ranging from QCIF (176x144) up to 1080p (1920x1080) is used in this evaluation. Moreover, the different compression parameters are experimentally fine-tuned to obtain the best compression efficiency for each codec, which is not the case of evaluations described in the literature.

Further, the idea of a partially decoded frame is introduced in chapter 4. This frame is used to enhance the reconstruction of the finally decoded frame in a proposed improved SI technique. The partially decoded frame is obtained by DVC decoding using an initial SI. Then, suspicious motion vectors are detected using the residual between the initial SI and the partially decoded frame. Further, the corresponding motion vectors are re-estimated using bi-directional motion estimation and then smoothed. Finally, the reconstruction process is run for a second iteration with the re-estimated SI. As the difference in prediction quality between the initial and re-estimated SI increases, a better reconstruction enhancement is achieved. This is mainly the case for video with significant motion. Moreover, the same idea is used to propose a new hybrid EC for DVC. It uses spatial EC based on anisotropic diffusion [15] to improve the performance of temporal EC. More specifically, the partially decoded frame is obtained by applying spatial EC to the decoded frame. Then, the same process as in the improved SI (i.e. erroneous block detection, bi-directional motion estimation and smoothing) is used to perform temporal EC of the damaged blocks. The results show that the combination of both EC algorithms outperforms each one of them when applied separately (i.e. only spatial or temporal EC). The hybrid EC also outperforms error-concealed AVC/H.264 in its different coding modes.

DVC is also attractive for multiple camera scenarios since it implies separate encoding of the cameras views. Thus, a review of different SI techniques for multiview DVC, their prediction quality, complexity and RD performance are presented in chapter 5. Further, an iterative multiview SI technique for enhanced reconstruction, similar to the monoview improved SI, is also introduced in this chapter. The iterative SI uses an initial SI, which depends on the amount of motion in the video, to initially decode the WZ frame (i.e. turbo decoding and reconstruction). Then, a better SI is generated by performing motion estimation and compensation as in conventional coding using the initially decoded

WZ frame and four reference frames: the previous frame, the forward frame, the left camera frame and the right camera frame. Finally, the reconstruction process is run again with the new SI to generate a more enhanced version of the WZ frame. This is shown to be efficient for video with high activity whereas the gain is negligible for low motion video. In addition to the iterative SI, two fusion techniques combining the least correlated side informations are proposed to improve the performance over monoview DVC in a multiview scenario. In other words, the results show that there is a significant gain in performance by exploiting the inter-view correlation in addition to the intra-view one.

Codec-independent scalability is an interesting feature, which can be enabled by DVC. This implies independency between the base and the enhancement layer as DVC is based on statistical framework. In chapter 6, scalable schemes for image and video coding based on DVC are introduced. They address temporal, spatial as well as quality scalability. This is achieved by conventional encoding of the base layer and generating the corresponding enhancement layers as WZ bits. In this case, the SI is generated by either temporal motion estimation and compensation or spatial upsampling. Further, these schemes are evaluated in error-free and error-prone environments. In the latter, the DVC-based schemes show very good robustness with respect to errors even at high error rates, which is not the case of conventional coding. For the latter, a Reed Solomon (RS) [16] protection scheme is applied with different protection strengths, weak, average and high. For the first two, the cliff effect is observed beyond a certain rate as a significant drop in performance is noticed. For high RS protection, the overhead of parity bits is large enough to correct all the errors but the conventional coding performance drops such that it is outperformed by the scalable DVC schemes.

Further, the issue of privacy is addressed for DVC by efficient transform domain scrambling in chapter 7. The goal is to conceal information within regions of interest in video for privacy sensitive applications such as video surveillance. At the same time, the scheme has to enable unscrambling only for authorized users to visualize the video in a clear version. The scrambling consists in either pseudo-randomly inverting or permuting the transform coefficients in the regions of interest. Both scrambling techniques are driven by a pseudo-random generator, which is initialized by a seed. The latter constitutes the secret key, which is required at the decoder to undo the scrambling. The permutation scrambling is more secure than the one based on sign inverting as the latter requires a lower complexity brute force attack. On the other hand, the sign inverting scrambling results in a lower rate increase than the permutation scrambling with respect to the original DVC scheme without scrambling.

Then, the conception and the implementation of a DVC demonstrator based on the DISCOVER software [17] is described in chapter 8. This demonstrator is defined in a setup being as close as possible to a complete real application scenario based only on realistic assumptions, where software complexity, hardware requirements, network transmission and practical deployment aspects are taken into account. Moreover, a careful study of the potential limiting factors of the DVC software is presented, namely the real time encoding, the presence of a feedback channel and the real time decoding. The latter is the target of an exhaustive algorithm optimization effort realized on the DVC decoder software, which is also described in this chapter.

Finally, chapter 9 ends this thesis and draws the main conclusions about this work in addition to future work.

Chapter 2

Distributed Video Coding (DVC)

2.1 Introduction

The foundations of DVC go back to the 70's as Slepian and Wolf (SW) [3] established the achievable rates for lossless coding of two correlated sources in different configurations. Then, Wyner and Ziv (WZ) [4] extended the SW theorem to the lossy case. It was until lately that the first practical implementations of DVC were introduced in [5, 18].

The DVC theoretical foundations as well as different practical implementations of DVC are described in this chapter, with an emphasis on the scheme used in this research. Unlike conventional encoders (e.g. AVC/H.264 [13]), where the source statistics are exploited at the encoder side, DVC can shift this task towards the decoder side. This would result in encoders, which are low in terms of complexity. On the other hand, DVC decoders would be highly complex in this case. Therefore, DVC is suitable for some emerging applications, where computational power is sparse at the encoder side such as wireless low power video surveillance, multimedia sensor networks, wireless PC cameras and mobile camera phone. Furthermore, DVC is based on a statistical framework, not a deterministic one, that makes it have good error resilience properties [19, 20]. DVC can be used to design codec independent scalable codecs as in [21]. In other words, the enhancement layer is independent from the base layer codec.

This chapter is organized as follows. First, the theoretical foundations of DVC are reviewed in section 2.2. Then, some practical implementations of DVC from the literature are described in section 2.3. Further, the practical DVC scheme used in this research is described in details in section 2.4. An evaluation of DVC's performance in error-free and error-prone environments is presented in section 2.5. Section 2.6 identifies some applications for which DVC would be attractive and beneficial. Finally, section 2.7 concludes the chapter.

2.2 Theoretical DVC

SW [3] defined the Distributed Source Coding (DSC) problem of coding correlated sources as illustrated in Figure 2.1. In the latter, X and Y are two statistically dependent random sequences and each switch S_i with $i \in \{1, 2, 3, 4\}$ is closed if its value is equal to 1 otherwise it is open.

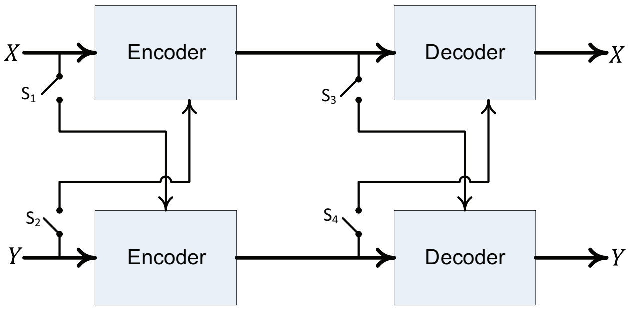


Figure 2.1: The different cases of correlated source coding. The switches S_i with $i \in \{1, 2, 3, 4\}$ are either open (i.e. equal to 1) or closed (i.e. equal to 0) [3].

For example, the state (1111) (i.e. all switches are closed) refers to joint encoding and joint decoding of the correlated sources. In other words, the switches define whether the source's encoder/decoder has access to the other source. SW established the admissible rates for each state of the switches. The case (0011) represents by far the most interesting result in terms of novelty presented by SW in [3]. This case corresponds to separate encoding and joint decoding of X and Y and its admissible rate region is depicted in Figure 2.2, which is defined by

$$\begin{aligned} &R_X \geq H(X|Y),\ R_Y \geq H(Y|X), \\ &R_X + R_Y \geq H(X,Y). \end{aligned} \tag{2.1}$$

Despite the separate encoding of X and Y, SW proves that the total rate ,$R_X + R_Y$, for encoding X and Y can achieve the joint entropy $H(X,Y)$ as if the they were jointly encoded. Decoding with SI (Figure 2.3) is considered as a special case of DSC. In this case, the source X depends on some SI Y. A rate $R_X \geq H(Y)$ can be achieved by entropy encoding to transmit Y to the decoder. For X, a rate $R_X \geq H(X|Y)$ is admissible. Therefore, the total rate is $R_X + R_Y \geq H(Y) + H(X|Y)$, which corresponds to a point on the region's border depicted in Figure 2.2. This stands whether SI Y is available at the encoder side or not.

Later on, WZ [4] established the bounds for lossy compression as an extension to the SW theorem, where the decoder produces $\hat{X}$ with a certain distortion D with respect to X as illustrated in Figure 2.4. When the SI is available at both, encoder and decoder sides, a rate $R_{X|Y}(D)$ is achieved for encoding X with a distortion D. Further, there is an increase of $(R^{WZ}_{X|Y}(D) - R_{X|Y}(D)) \geq 0$ in rate when the SI is not available at the encoder but only at the decoder side. In other words, the rate in the case where the SI is not available at the encoder is lower bounded by the one when the SI is available at the encoder. Nevertheless, WZ show that both rates, $R^{WZ}_{X|Y}(D)$ and $R_{X|Y}(D)$), are equal when the sources

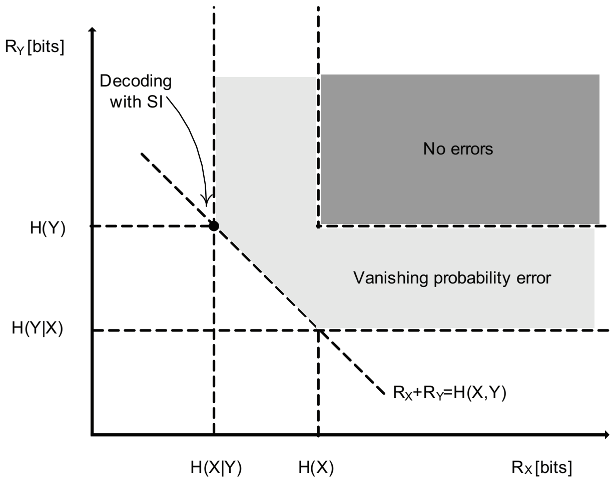


Figure 2.2: Achievable rate region defined by the Slepian-Wolf bounds for the switches case (0011).

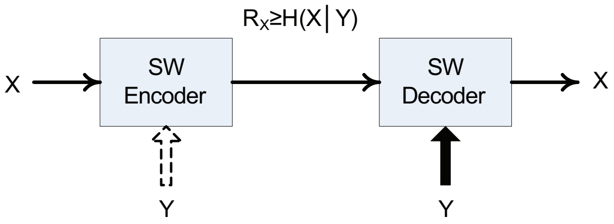


Figure 2.3: Lossless DSC of source X, which depends on SI Y [18].

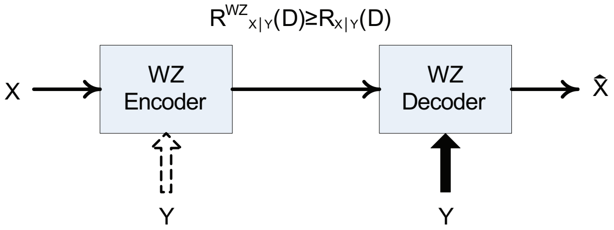


Figure 2.4: Lossy DSC of source X, which depends on SI Y [18].

are memoryless Gaussian and the Mean Square Error (MSE) is used as the distortion metric.

2.3 DVC Implementations

2.3.1 PRISM Codec

PRISM is DVC's practical implementation by the Berkeley's group introduced in [5]. Initially, the input video is divided into Group Of Pictures (GOPs). Each one consisits of the frames on which a complete cycle of encoding and decoding process is applied.

The first frame of a GOP is encoded using a traditional technique such as AVC/H.264 Intra [13]. This frame is then used as a starting point for encoding the remaining frames of the GOP. They are processed using a hybrid technique, which combines distributed and traditional coding.

First, each frame is split into 8x8 blocks and then transformed, where each block is considered as a separate unit and is independently encoded from its spatially neighboring blocks. Then, the encoder estimates the current block's correlation level with the previous frame by using a zero-motion block matching.

Further, the blocks are classified into different encoding classes depending on the level of estimated correlation (Figure 2.5). The first class of blocks is the ones with very low correlation, which are encoded using a conventional coding approach. Blocks with very high correlation are signaled as *skip* blocks, which are not encoded and constitute the second class. Finally, the remaining blocks with in-between correlation are encoded using a distributed approach, where the used number of bits depends on the estimated correlation level. More specifically, the encoder computes syndrome bits [7, 8, 22] for the transformed coefficients of the block as well as a 16-bit Cyclic Redundancy Check (CRC).

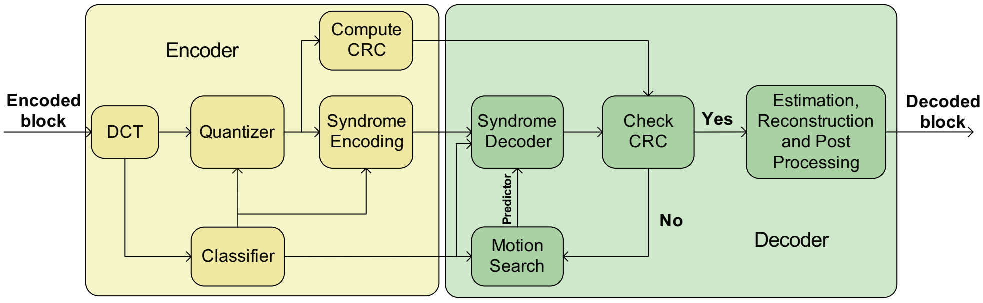


Figure 2.5: PRISM codec architecture [23].

At the decoder, the syndrome bits are used to correct different predictors. Then, the CRC is used to check if the decoding is successful.

2.3.2 The Stanford Codec

As for the PRISM codec, the video sequence is split into different GOPs in the Stanford codec [9]. Similarly, the first frame of every GOP is encoded using a traditional encoding technique and is called a key frame. The remaining frames are considered as WZ frames and they are completely encoded in a distributed fashion as shown in Figure 2.6. This means that the whole content of the frame is encoded using DSC techniques as turbo codes [10] are used to extract parity bits from each WZ frame.

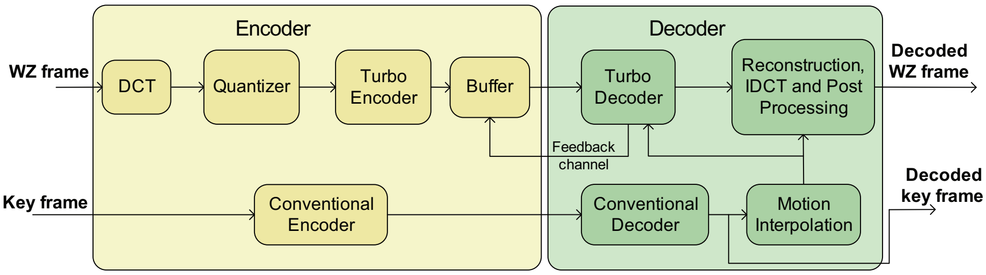


Figure 2.6: Stanford codec architecture [23].

At the decoder, the Stanford's approach proposes to use the key frames to estimate the motion and then constructs a prediction (i.e. SI) of the WZ frame by interpolation between key frames. The WZ decoder corrects this prediction to obtain a higher quality approximation of the WZ frame using parity bits received from the encoder upon request via the feedback channel.
It is obvious that the most important difference with respect to the PRISM codec is the fact that the WZ frame is encoded as a whole and the motion is completely estimated before starting the WZ decoding. Furthermore, the presence of a feedback channel also differentiates the Stanford codec from PRISM.

2.3.3 DVC Tools

Most of DVC research in the literature is based on these both schemes, PRISM and the Stanford approach. Moreover, different tools were proposed to improve the performance of DVC as it still inferior to conventional coding in trem of compression efficiency. The efficiency of DVC depends on how much correlated is the SI with the original WZ data. Since linear interpolation of the motion field between key frames is adopted to generate SI, the latter is poorly correlated with the WZ frame. A better estimation is achieved by removing motion discontinuities at the boundaries and outliers in homogeneous regions [24] by spatial smoothing and refinement of the interpolated motion vectors between the key frames. Further, occlusion regions are dealt with by forward and backward extrapolation of the key frames in [25]. More specifically, two side informations are constructed by applying forward and backward motion vectors to the previous and the forward key frame, respectively. Instead of averaging both side informations as most schemes do (e.g. [24]) and use the final SI in the decoding process, both side informations are used simultaneously in the decoding process. The encoder could send a priori information on the WZ frames to help the decoder in constructing a better SI. This information could be CRCs as in [5] or hash codes as in [26,27]. More specifically, the decoder has access to different SI block candidates and verifies whether the decoded CRC (or hash) matches the one received from the encoder. However, this approach requires multiple WZ decoding steps, which increases the decoder complexity, and implies a transmission rate overhead.

In [28], feature points extracted in the WZ frames are transmitted as additional information to help correcting misalignments in 3D model-based frame interpolation. The latter accounts for scene's geometric constraints and significantly improves the SI quality for static scenes captured by a moving camera.

The performance of DVC is strongly related to its capability to estimate the correlation model between SI and WZ frames. Initially, a Laplacian model is computed offline for each sequence. This assumes that the originals are available at the decoder, which is not feasible in practice. Then, a technique for online estimation (i.e. not requiring the originals) of the correlation model at the decoder is introduced in [29]. Alternatively, the encoder can derive the correlation model parameters by estimating the SI at the encoder [30]. On the other hand, this would increase the encoder's complexity.

The rate allocation in DVC depends on the correlations between SI and the original data. The rate is defined by the decoder as it requests parity bits from the encoder until successful decoding or all the bits are exhausted. Further, a hybrid rate control is proposed in [31], where the encoder initially estimates the amount of parity bits to send to the decoder. If the error rate at the decoder is still significant, the decoder request additional parity bits via the feedback channel. The initial rate sent by the encoder limits the use of the feedback channel leading to lower delay and decoder's complexity. Further, a practical stopping criteria is proposed in [32] by estimating the optimal number of parity bits requests required for successful decoding.

Finally, due the low correlation between SI and the WZ data in regions of occlusion, a block-based coding mode selection by estimating the SI at the encoder side is introduced in [5, 33]. The zero-motion compensated block from the previous frame is used to estimate the correlation level for the current block, which defines its encoding mode as explained previously for PRISM. In a similar way, a block-basis mode decision mechanism based on the estimated correlation is proposed in [33]. The Intra coding mode is activated for a block in the case of a weak correlation estimation. This is assumed true for blocks with relatively small temporal prediction error and low luminance variance. This means that the block is smooth, which is in favor of Intra coding as it efficiently exploits spatial redundancy.

In [34], an analysis of the coding efficiency of DVC schemes that perform motion interpolation at the decoder side is presented. The precision of the estimation depends mainly on the motion field's temporal coherence and the distance between the successive key frames. Moreover, a model, based on a state-space model and Kalman filtering, is proposed to evaluate the overall performance of the DVC codec. The model shows that DVC, based on motion interpolation at the decoder, cannot achieve the performance of predictive convectional coding. It also estimates the optimum GOP size that provides the best performance for the DVC scheme.

2.4 Practical DVC

Figure 2.7 shows the DVC architecture [17,35,36] used in this research. Initially, the different entities on the encoder side are described with an emphasis on the WZ encoder. Further, the critical blocks on the decoder side are also presented.

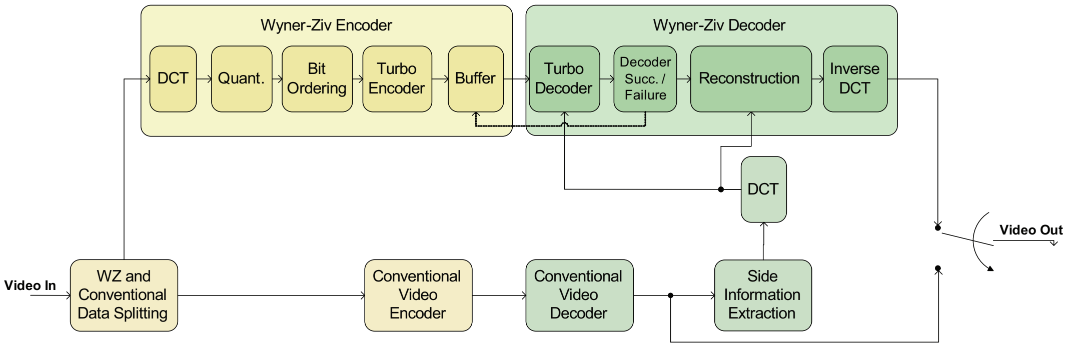


Figure 2.7: Conventional DVC architecture [17].

2.4.1 DVC Encoder

At the encoder, the frames are split into two sets. The first one is the key frames, which are fed to a conventional video encoder such as AVC/H.264 [13] Intra. Since chapter 3 is dedicated to Intra frame encoding, the latter is not described in this chapter. The other set is the WZ frames. The latter are transformed and then quantized with a quantization level that depends on the desired quality. The same 4x4 separable integer transform as in AVC/H.264 is used with properties similar to the Discrete Cosine Transform (DCT) [37]. Then, the quantized coefficients are organized in bands and then separated into bit planes, which are fed one by one to a turbo encoder [10]. The latter offers near-channel capacity error correcting capability. Furthermore, a CRC [38] is computed for each quantized bitplane and transmitted to the decoder. All generated parity bits are stored at the encoder side in a buffer even if only a subset is used in the decoding phase upon the decoder's request via the feedback channel.

Transformation

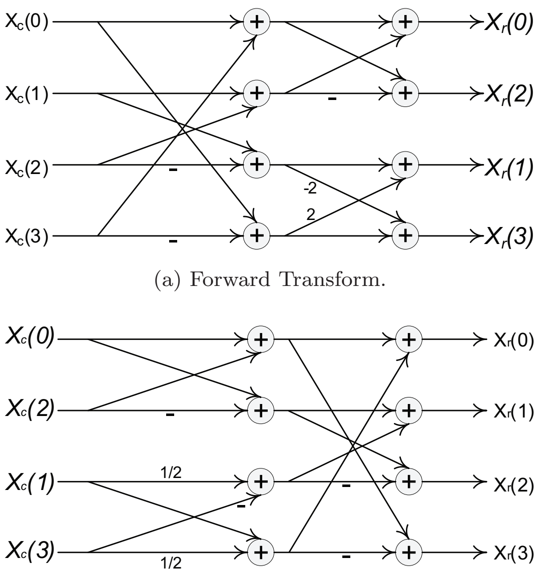
The aim of the transformation phase is to make the input video more suitable for compression by compacting the signal's energy into the lower transform coefficients. The DVC scheme uses the 4x4 separable integer transform in AVC/H.264 with properties similar to the DCT [37]. The forward transform is defined as follows

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}. \tag{2.2}$$

The inverse transform is given by

$$H_{inv} = \begin{pmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{pmatrix}. \tag{2.3}$$

Both forward and reverse transforms are multiply-free as they can be implemented with only additions and shifts. Figure 2.8 depicts the implementation



(a) Forward Transform.

(b) Reverse Transform.

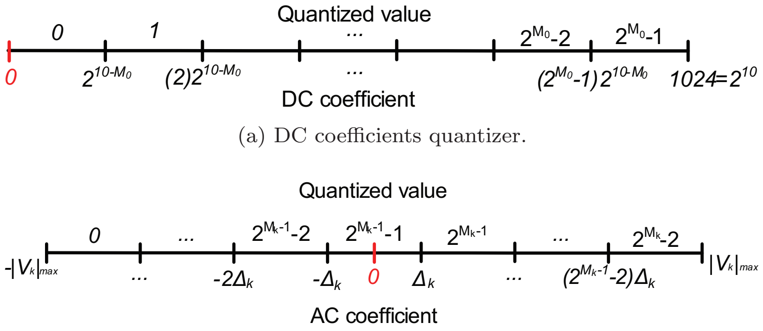
Figure 2.8: Implementation of the forward and reverse transforms [35].

of both transformations that maps the column elements in the input to row elements in the output.

Quantization

Quantization is essential for lossy compression, which does not imply perfect source reconstruction. It allows for high compression efficiency without negatively impacting the visual quality. Two different quantization approaches are used depending the quantized coefficient:

- The DC coefficient represents the average energy of the transformed block. Thus, the DC coefficient takes large positive values. Therefore, it is quantized using the uniform scalar quantizer shown in Figure 2.9 (a).



(a) DC coefficients quantizer.

(b) AC coefficients quantizer.

Figure 2.9: Quantization of the different bands [35].

- The AC coefficients are mainly concentrated around zero. Therefore, they are quantized using a uniform scalar quantizer with a symmetric interval around zero as depicted in Figure 2.9 (b). Thus, low AC coefficient around zero are quantized under the same quantization interval index independently of its sign. This would avoid errors between quantized symbols of the SI and the original frame. Otherwise, low AC coefficients with opposite signs would be quantized to different indexes if a quantizer without a symmetric interval around zero is used. Moreover, the turbo decoder would try to correct this mismatch. If it fails, blocking artifacts would appear in the decoded frame.

A quantization step is assigned to each frequency band depending on its range and importance. The smaller the quantization step size, the lower is the distortion at the decoder. For 8 bit luminance samples, the DC band's dynamic range is 1024. Thus, the quantization step size for the DC band b_0 is defined as

$$\Delta_0 = 2^{10-M_0} = \frac{1024}{2^{M_0}}, \tag{2.4}$$

where 2_0^M is the number of the quantization levels for the DC band. For the AC bands b_k $k \in \{1, ..., 15\}$, the quantization step size Δ_k is computed as

$$\Delta_k = \frac{2|V_k|_{max}}{2^{M_k} - 1}, \tag{2.5}$$

where $|V_k|_{max}$ stands for the highest absolute value within b_k and 2^{M_k} stands for the number of quantization levels.
By varying M_k for the different bands ,b_k, the quality of decoded WZ frame is changed. In this research, 8 RD points are possible for the WZ frames corresponding to the following 8 Quantization Indexes (QI)

$$QI_1 = \begin{pmatrix} 16 & 8 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, QI_2 = \begin{pmatrix} 32 & 8 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, QI_3 = \begin{pmatrix} 32 & 8 & 4 & 0 \\ 8 & 4 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$QI_4 = \begin{pmatrix} 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 0 \\ 8 & 4 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{pmatrix}, QI_5 = \begin{pmatrix} 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 4 \\ 8 & 4 & 4 & 0 \\ 4 & 4 & 0 & 0 \end{pmatrix}, QI_6 = \begin{pmatrix} 64 & 16 & 8 & 8 \\ 16 & 8 & 8 & 4 \\ 8 & 8 & 4 & 4 \\ 8 & 4 & 4 & 0 \end{pmatrix},$$

$$QI_7 = \begin{pmatrix} 64 & 32 & 16 & 8 \\ 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 4 \\ 8 & 4 & 4 & 0 \end{pmatrix}, QI_8 = \begin{pmatrix} 128 & 64 & 32 & 16 \\ 64 & 32 & 16 & 8 \\ 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 0 \end{pmatrix}.$$

Each element in a matrix defines the number of quantization levels for the corresponding frequency band. A 0 in a matrix means that no WZ bits are sent for the corresponding frequency band and the corresponding coefficients from the SI are used in the finally decoded WZ frame.
The different quantized coefficients of the same band are grouped together and the different bit planes are extracted. The latter are organized from the most significant, $M_k - 1$, to the least significant one, 0.

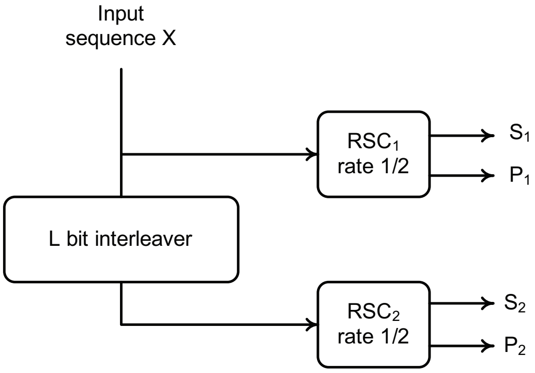
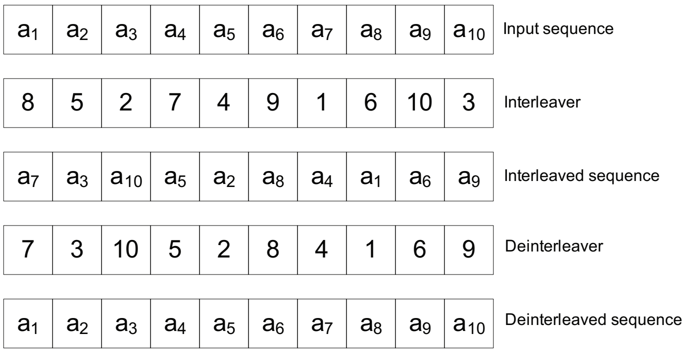


Figure 2.10: Turbo encoder's structure [35].

Turbo Encoder

Turbo codes [10] were introduced by Berrou et al. in the field of digital communications as they offer near-channel capacity error correcting capabilities. They consist in a Parallel Concatenation of Recursive Systematic Convolutional Codes (PC-RSC) in addition to an interleaver to spread burst errors as depicted in Figure 2.10. Each RSC_i encoder produces two outputs, the systematic output, S_i, and the parity output, P_i, where $i \in \{1, 2\}$.



a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	Input sequence
8	5	2	7	4	9	1	6	10	3	Interleaver
a_7	a_3	a_{10}	a_5	a_2	a_8	a_4	a_1	a_6	a_9	Interleaved sequence
7	3	10	5	2	8	4	1	6	9	Deinterleaver
a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	Deinterleaved sequence

Figure 2.11: Interleaving and deinterleaving of a 10-bit sequence [35].

First, one of the turbo encoder's architectural modules, the interleaver, is described. The interleaver's output sequence is its input sequence rearranged in a different order following a given pattern. The sequence's original order is recovered by feeding the interleaved sequence into a deinterleaver. Figure 2.11 illustrates the interleaving of a 10-bit length sequence. The input bit located at the 1^{st} position, a_1, is mapped to the 8^{th} position in the output sequence. The 2^{nd} input bit, a_2, is mapped to the 5^th position in the output sequence and so on.

The deinterleaving structure is defined as follows

$$deinterleaver[interleaver[j]] = j, \tag{2.6}$$

where the value of the *deinterleaver* array at the position $interleaver[j]$ is equal to j.

In this research, a pseudo-random interleaver is used as defined in [39]. Moreover, the turbo coding performance depends on the interleaver's length L, which must have a large value. The reason for this is that low values for L might result in a lack of randomness that would compromise the performance of the turbo encoder.

The turbo encoder has two RSC encoders, where each one is characterized by a generator matrix G such that the encoder's output, RSC_{out}, is computed as the product of the matrix G and the encoder's input, RSC_{in}.

$$RSC_{out} = RSC_{in}.G. \tag{2.7}$$

The generator matrix G is defined such that

$$G = \begin{pmatrix} 1 & \frac{g_2(D)}{g_1(D)} \end{pmatrix}, \tag{2.8}$$

where $g_1(D)$ and $g_2(D)$ are two polynomials defined as $g_i(D) = g_{i0} + g_{i1}D + g_{i2}D^2 + ... + g_{im}D^m$, $i = 1, 2$ and D denotes a delay.

The polynomials degree m corresponds to the memory of the RSC encoder, which corresponds to the number of register shift elements D used in implementing the RSC encoder. The coefficients g_{ik} (i = 1, 2 and k = 1, 2, ..., m) take the value 1 or 0 indicating whether the shift register D^k is retained or not when computing RSC_i encoder's output. Moreover, the number of states of the RSC encoder is equal to 2^m since only two values are possible for each g_{ik} coefficient. The encoder's rate is defined as the ratio of number of bits at the input over the number of bits at the output.

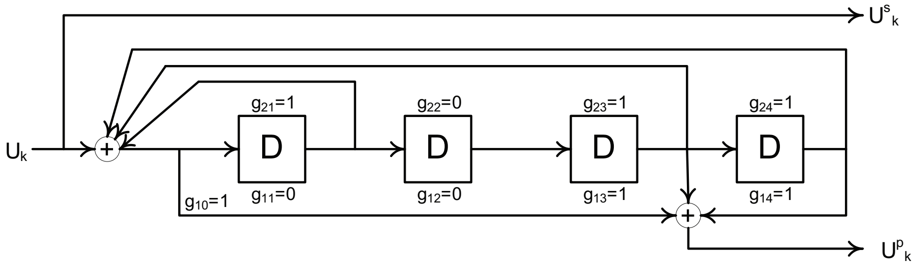


Figure 2.12: Rate $\frac{1}{2}$ (one input, two output) RSC encoder with memory 4 (16 states) and generator matrix given by equation 2.9 [35].

Figure 2.12 depicts the RSC encoder used in this research. It has a rate of $\frac{1}{2}$, a memory, m, equal to 4 and a generator matrix given by

$$G = \begin{pmatrix} 1 & \frac{1+D+D^3+D^4}{1+D^3+D^4} \end{pmatrix}. \tag{2.9}$$

The symbol u_k represents the k^{th} bit of the input L-bit sequence. The symbols u_k^s and u_k^p are the outputs of the RSC encoder, the systematic and the parity bit, respectively. The systematic bit is a copy of the input bit, which is the reason the encoder is called systematic. The parity bit, u_k^p, for a 16 state RSC code is computed as

$$u_k^p = g_{10} + (s_1^{k-1}.g_{11} + s_2^{k-1}.g_{12} + s_3^{k-1}.g_{13} + s_4^{k-1}.g_{14}), \tag{2.10}$$

where $g_{10} = u_k + (s_1^{k-1}.g_{21} + s_2^{k-1}.g_{22} + s_3^{k-1}.g_{23} + s_4^{k-1}.g_{24})$ and the operator (.) is the exclusive OR operator. Coefficients s_1^{k-1} to s_4^{k-1} correspond to the shift registers values at time $k-1$. Moreover, the set of values $S_{k-1} = (s_1, s_2, s_3, s_4)^{k-1}$ is called the RSC encoder's state S at time $k-1$. In this context, the term time refers to the position within the input L-bit sequence. After introducing the k^{th} input bit, the shift registers values change depending on the input bit with a state transition from S_{k-1} to S_k.

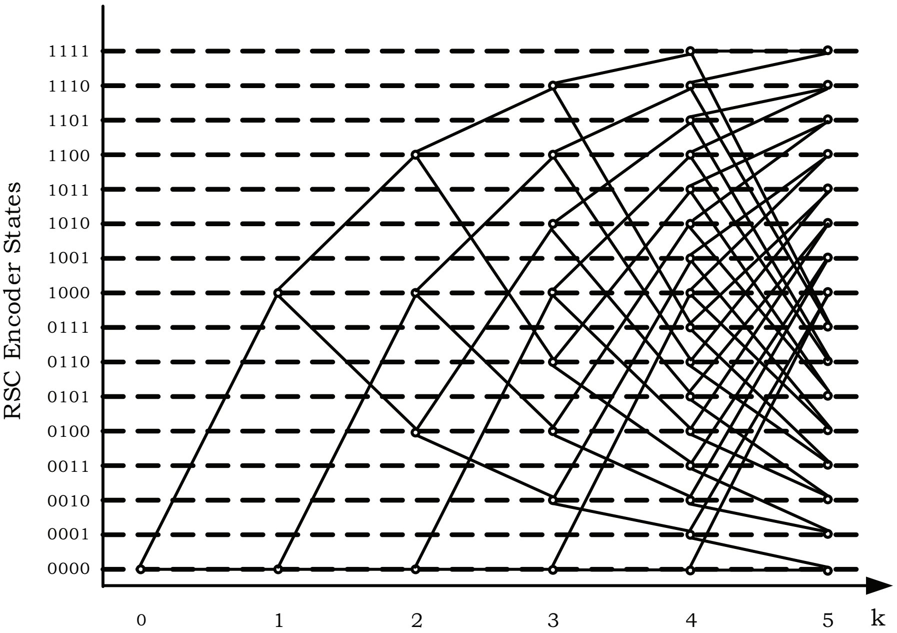


Figure 2.13: Trellis diagram of a RSC encoder with a generator matrix given by equation 2.9. Each state is represented by a circle and each state transition is represented by a line connecting two circles (RSC states) [35].

Figure 2.13 depicts the trellis diagram [40] of a RSC encoder with a generator matrix given by equation 2.9 and the initial state being 0000. Since the polynomial's degree, m, is equal to 4, there are $2^4 = 16$ states.

Shifting the first bit of the L-bit input sequence, u_1, into the RSC encoder, two state transitions are possible, the ones associated with $u_1 = 0$ and $u_1 = 1$. For each u_1 bit value, 0 or 1, corresponds an output sequence formed by u_1^s and u_1^p. Further, two state transitions are possible for each state resulting from shifting u_2 into the RSC encoder and so on.

The different states and the corresponding input and output bits can be also defined as in Table 2.1.

Each one of the two RSC encoders included in the turbo encoder computes a parity sequence corresponding to the L-bit sequence at its input. For RSC_2, the L-bit input sequence is a pseudo-randomly interleaved version of the input at RSC_1. The outputted systematic bits, S_1 and S_2, are discarded while, the parity sequences, P_1 and P_2, are stored in a buffer at the encoder.

The WZ codec is built based on a Rate Compatible Punctured Turbo (RCPT) [41] code structure as defined in [39]. More specifically, the generated parity

Table 2.1: Possible RSC encoder's state transitions, $S_{k-1} \rightarrow S_k$, and output bits, (u_k^s, u_k^p), given the RSC encoder's input bit u_k.

State S_{k-1}	u_k	u_k^s	u_k^p	States S_k
0000	0/1	0/1	0/1	0000/1000
0001	0/1	0/1	0/1	1000/0000
0010	0/1	0/1	0/1	1001/0001
0011	0/1	0/1	0/1	0001/1001
0100	0/1	0/1	0/1	0010/1010
0101	0/1	0/1	0/1	1010/0010
0110	0/1	0/1	0/1	1011/0011
0111	0/1	0/1	0/1	0011/1011
1000	0/1	0/1	1/0	1100/0100
1001	0/1	0/1	1/0	0100/1100
1010	0/1	0/1	1/0	0101/1101
1011	0/1	0/1	1/0	1101/0101
1100	0/1	0/1	1/0	1110/0110
1101	0/1	0/1	1/0	0110/1110
1110	0/1	0/1	1/0	0111/1111
1111	0/1	0/1	1/0	1111/0111

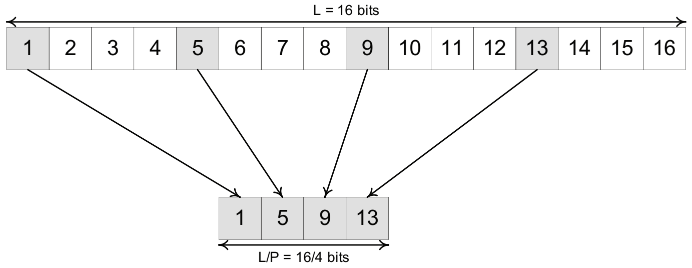


Figure 2.14: 16-bit parity sequence division process considering a puncturing period of 4 (L=16,P=4) [35].

sequence is divided into P blocks of (L/P) bits each, where P is the puncturing period. Figure 2.14 illustrates a parity sequence division process.

The first block of (16/4) consists of the parity bits located at positions 1, 5, 9 and 13. The bits at positions 2, 6, 10 and 14 form the second block of (16/4) bits and so on. In this research, the puncturing period, P, is equal to 48. Moreover, a pseudo-random puncturing pattern is used, which allows to achieve good RD performance. In other words, the blocks transmission is performed in a pseudo-random way as the blocks are transmitted alternately from P_1 and P_2 except for the first request. In the latter, the WZ encoder sends two (L/P) bit blocks, one from each parity sequence P_1 and P_2. This process is carried out until no more requests are made or all the parity bits have been transmitted.

2.4.2 DVC Decoder

At the decoder, the key frames are initially decoded. Then, Motion Compensation Temporal Interpolation (MCTI) is used to generate SI for the WZ decoder by temporally interpolating the key frames. A virtual channel is used to model the correlation between the DCT coefficients of the original and SI frames. It is shown that the residual of the DCT coefficients follows a Laplacian distribution [18]. Then, turbo decoding is run using the SI and the virtual channel model to recover the correct WZ DCT bins. Finally, the reconstruction process [9] uses the SI along with decoded bins to recover the original frame up to a certain quality. This DVC scheme is decoder driven as the request for parity bits from the encoder is performed via a feedback channel until the decoding is successful or the parity bits are exhausted. The decoding is considered successful if the decoded bitplane's error probability is lower than 10^{-3} and its CRC matches the one received from the encoder.

Side Information Generation

The SI, which is an estimation of the WZ frame at the decoder, is computed by MCTI. MCTI uses the key frames to perform motion estimation. The resulting motion vectors are interpolated at midpoint as illustrated in Figure 2.15.

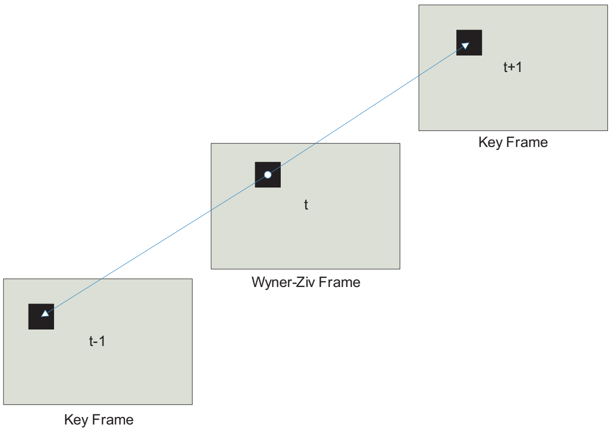


Figure 2.15: Motion Compensation Temporal Interpolation.

Each interpolated block is assigned the motion vector intercepting the block and being the closest to its center. Then, it is refined by preforming motion search in a small window on the linear trajectory between the next and the previous key frames and passing through the center of the interpolated block as shown in Figure 2.16.

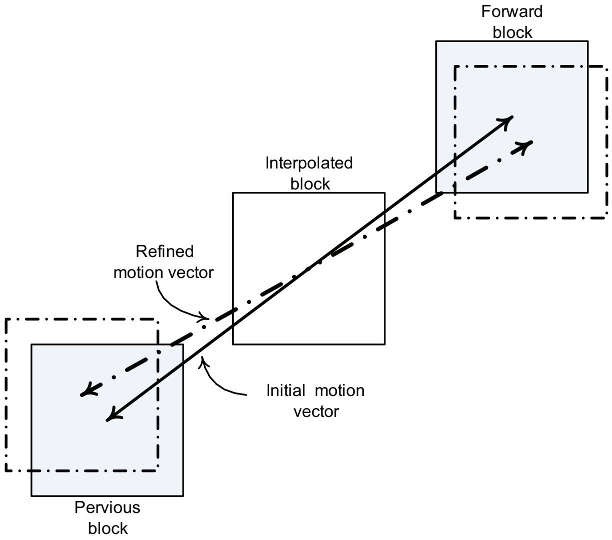


Figure 2.16: Motion refinement by bi-directional motion estimation [24].

Then, the obtained bi-directional motion vectors are smoothed using the filter defined in [24]. The resulting motion vector, mv_{wmf},is chosen in order to minimize the sum of distances (L^2-norm) to the neighboring blocks motion vectors as defined by Equation 2.11.

$$\sum_{j=1}^{N} w_j ||mv_{wmf} - mv_j||_L \leq \sum_{j=1}^{N} w_j ||mv_i - mv_j||_L, \tag{2.11}$$

where mv_1, ..., mv_N are the motion vectors of the current block and its neighboring blocks. The weights, w_j, are computed as follows

$$w_j = \frac{MSE(mv_c, B)}{MSE(mv_j, B)}, \tag{2.12}$$

where mv_c is the candidate motion vector for the current block, B, to be smoothed and MSE is the Mean Square Error between the key frames for the current block with the corresponding motion vector. According to Equations 2.11 and 2.12, the block is smoothed such that the re-estimated motion vector minimizes the prediction error and maintains the spatial coherence of the motion field.

Depending on the application requirements, different GOP sizes can be applied. The algorithm for SI generation works in the same way starting from adjacent already decoded frames. In Figure 2.17, an example of GOP size 4 is depicted. First, the SI of the middle frame is generated using both temporarily adjacent key frames. After the middle frame is decoded, it is regarded as a key frame and the procedure is repeated until all WZ frames are decoded within the GOP.

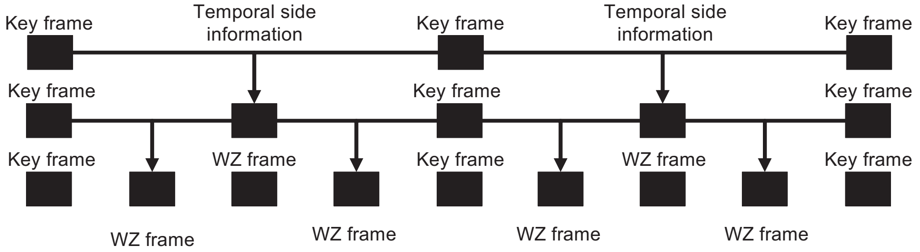


Figure 2.17: Example of GOP interpolation order of size 4.

Virtual Channel Model

The virtual channel model captures the correlation between the transformed original data and the SI at the decoder. After study of many video sequences, it is noted many times in the literature [18] that the difference between the original DCT coefficients and the transformed SI follows a Laplacian distribution. The latter has the following Probability Density Function (PDF)

$$PDF(I_{org} - I_{SI}) = \frac{\alpha}{2} e^{|I_{org} - I_{SI}|}, \tag{2.13}$$

where I_{org} is the original value of a given band coefficient in the original sequence, I_{SI} is the value of that same band coefficient in the SI and α is a parameter that controls the variance of the distribution (i.e. how much the SI resembles the original data)

$$\alpha = \frac{2}{\sigma^2}. \tag{2.14}$$

The α parameter of the distribution in equation 2.13 is estimated by computing the variance of the residual between WZ and SI frames offline. Then, equation 2.14 is used to calculate α from the variance.
However, computing the alpha parameter is not realistic and feasible in a practical scenario as this would require the originals at the decoder. An online estimation of this parameter is possible as presented in [29]. First, the residual frame R, between the motion compensated versions of the previous and the forward key frames, X_p and X_f, as follows

$$R = abs(DCT(X_p) - DCT(X_f)). \tag{2.15}$$

Then, the variance σ_R^2 of R is then calculated as

$$\sigma_R^2 = E(R^2) - E(R)^2. \tag{2.16}$$

σ_R^2 is considered as a confidence measure of the SI creation process. Ideally, σ_R^2 should be close to σ^2, the variance of the residual between the WZ and the SI frames. Finally, Equation 2.14 is used to compute α by substituting σ^2 with σ_R^2.

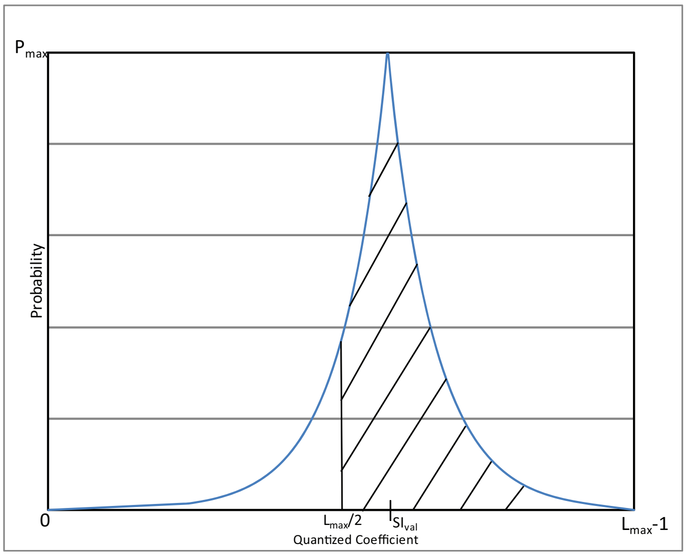


Figure 2.18: Conversion from PDF to bit probability.

To convert this distribution into bit probability, the PDF in equation 2.13 is integrated inside the interval corresponding to that bit being 1. For example, Figure 2.18 shows the probability a DCT coefficient value has of being the true original value, given that the SI value is SI_{val}. Thus, values around SI_{val} are very likely whereas values far from SI_{val} are very unlikely. To compute the probability that the most significant bit is 1, the shaded area is integrated and normalized. This interval corresponds to the most significant bit being 1 (i.e Values from $L_{max}/2$ to $L_{max}-1$).
To calculate the probabilities for the less significant bits, intervals are consecutively subdivided and each half is integrated and normalized. The obtained probabilities are used in the turbo decoder as channel probabilities, $P_{channel}$, as it is explained further.
A small example follows to explain how the intervals are determined for different bits being 1. Suppose that $L_{max}=8$, Table 2.2 shows the different possible values. To get the probability of the most significant bit being 1, the integration is made over the last four values (4 to 7). Then, the integration is made over the second half of the first and second intervals to compute the probability of the 2^{nd} bit being 1 (i.e. values 2,3 and 6,7) and so on.

Turbo Decoder

The turbo decoder architecture defined in [42] is used in this research. The turbo decoder is composed of two Soft Input Soft Output (SISO) decoders as depicted in Figure 2.19. P_1 and P_2 are the punctured versions of the parity bits produced by the turbo encoder. The systematic bits, S, are extracted directly from the SI, which can be seen as a corrupted version of the original data after passing through a virtual channel. The virtual channel model is used to try to predict the errors present in the SI.
Both SISO decoders exchange extrinsic information, which is a soft measure

Table 2.2: The different values for the quantized coefficient when $L_{max} = 8 = 2^3$.

	MSB is 1	2^{nd} bit is 1	LSB is 1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

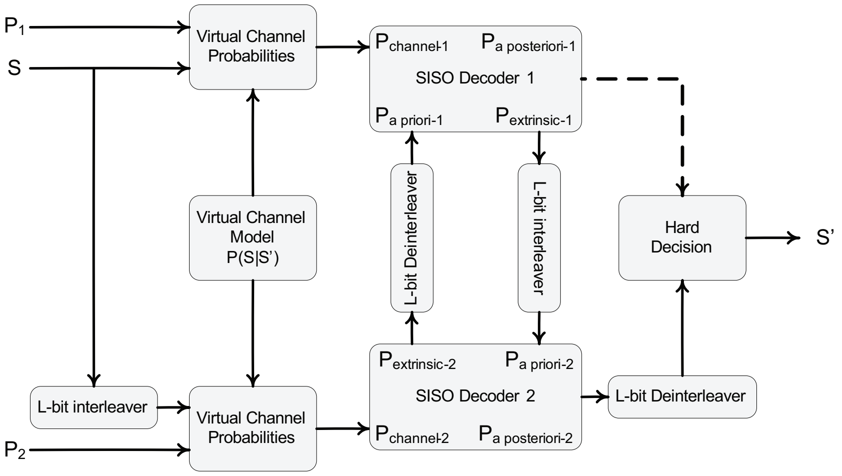


Figure 2.19: Diagram of the iterative turbo decoder, which uses two SISO decoders operating cooperatively [35].

of the confidence they have on the value of each bit. Furthermore, they use interleaved versions of the same data. The idea is that bursts of errors that would degrade the performance of one of the decoders are spread by the interleaver such that the other decoder's performance is not degraded as well. This is why all communication between the two SISO decoders is done via interleavering or deinterleavering. The hard decision module generates actual bits based on the obtained probabilities.

The SISO decoders require both channel probabilities and a priori information. Initially, SISO decoder 1 assumes that all bits have equal probabilities of being 0 or 1 as it does not have any a priori knowledge of the probabilities of the bits being decoded.

After decoding, SISO decoder 1 generates a vector of probabilities, $P_{aposteriori1}$. The latter states for each bit whether it is 0 or 1. It can also be used to take hard decisions on the bit values. Moreover, SISO decoder 1 produces an additional vector of probabilities, $P_{extrinsic1}$, which is used by SISO decoder 2 as a priori information as it is obtained using information, which is not available to SISO decoder 2. SISO decoder 2 is then executed and its $P_{extrinsic2}$ is fed into SISO decoder 1 restarting the whole process.

After a fixed number of iterations the process is stopped and the output of SISO decoder 2, $P_{aposteriori2}$, is used to take hard decisions. More specifically, when the probability of each bit being 0 is higher than the one of being 1, a 0 is decoded, otherwise a 1 is outputted. The output of SISO decoder 1, $P_{aposteriori1}$, can also be used in the hard decision module.

SISO decoders produce a soft output as they produce a real number measuring the decoder's confidence in the decoded bit being 0 or 1. This is opposite to a hard decoder, which directly outputs the actual decoded bits without a confidence measure.

In this research, the symbol-by-symbol Maximum A Posteriori algorithm (MAP) [43,44] is used as the SISO decoder. Let y_k^s and y_k^p be the systematic and parity bits available at the decoder, where k ranges from 0 to L. Assuming that there are no transmission errors, the parity bits, y_k^p, available at the decoder are exactly the same as the ones, u_k^p, generated by the encoder. The systematic bits, y_k^s, available at the decoder come from the SI and they are likely to be different from the ones, u_k^s, generated at the encoder.

Let the state of the encoder at time k be S_k. The bit d_k is associated with the transition from step $k-1$ to step k. The MAP algorithm computes the the Log-Likelihood Ratio (LLR or Λ), which is the logarithm of the ratio of the A Posteriori Probability (APP) of each information bit d_k being 1 to the APP of it being 0.

$$\Lambda(d_k) = ln\frac{\sum_{S_k}\sum_{S_{k-1}}\gamma_1(y_k^s, y_k^p, S_k, S_{k-1})\alpha_{k-1}(S_{k-1})\beta_k(S_k)}{\sum_{S_k}\sum_{S_{k-1}}\gamma_0(y_k^s, y_k^p, S_k, S_{k-1})\alpha_{k-1}(S_{k-1})\beta_k(S_k)}. \qquad (2.17)$$

The forward recursion of the MAP is expressed as

$$\alpha_k(S_k) = \frac{\sum_{S_{k-1}}\sum_{i=0}^{1}\gamma_1(y_k^s, y_k^p, S_k, S_{k-1})\alpha_{k-1}(S_{k-1})}{\sum_{S_k}\sum_{S_{k-1}}\sum_{i=0}^{1}\gamma_1(y_k^s, y_k^p, S_k, S_{k-1})\alpha_{k-1}(S_{k-1})}, \qquad (2.18)$$

$$\alpha_0(S_0) = \begin{cases} 1 & \text{if } S_0 = 0 \\ 0 & \text{Otherwise} \end{cases}. \qquad (2.19)$$

The backward recursion is defined as

$$\beta_k(S_k) = \frac{\sum_{S_{k+1}} \sum_{i=0}^{1} \gamma_1(y_{k+1}^s, y_{k+1}^p, S_k, S_{k+1}) \beta_{k+1}(S_{k+1})}{\sum_{S_k} \sum_{S_{k+1}} \sum_{i=0}^{1} \gamma_1(y_{k+1}^s, y_{k+1}^p, S_k, S_{k+1}) \beta_k(S_k)}, \tag{2.20}$$

$$\beta_L(S_L) = \begin{cases} 1 & \text{if } S_N = 0 \\ 0 & \text{Otherwise} \end{cases} . \tag{2.21}$$

The branch transition probabilities are given by

$$\begin{aligned} \gamma_i(y_k^s, y_k^p, S_k, S_{k-1}) &= q(d_k = i|S_k, S_{k-1}) p(y_k^s|d_k = i) \\ & \quad p(y_k^p|d_k = i, S_k, S_{k-1}) Pr(S_k|S_{k-1}). \end{aligned} \tag{2.22}$$

The value of $q(d_k = i|S_k, S_{k-1})$ is either 1 or 0 depending on whether bit i is associated with the transition from state S_{k-1} to S_k or not.

$$Pr(S_k|S_{k-1}) = \begin{cases} Pr(d_k = 1) & \text{if } q(d_k = 1|S_k, S_{k-1}) = 1 \\ Pr(d_k = 0) & \text{if } q(d_k = 0|S_k, S_{k-1}) = 1 \end{cases} . \tag{2.23}$$

After computing the α_k, β_k and γ_i terms, the final LLRs $\Lambda(dk)$ is evaluated. A positive LLR indicates that the bit should be 1 whereas a negative LLR indicates that it should be a 0. For more details, refer to [44].

Reconstruction

This block in the decoding process is opposite to the quantization step at the encoder. After turbo decoding, the decoder knows perfectly the quantization interval of each quantized value. Relying on the assumption that the WZ frame is correlated with the SI, the reconstruction block uses the SI along with decoded bins to improve reconstruction quality as described in [9]. The principal consists in either accepting a SI value as a reconstructed value if it fits into the quantization interval corresponding to the decoded bin or truncating the SI value into this quantization interval. The reconstruction is performed independently for every transform coefficient of every band.

Let Y be the SI value, d the decoded quantized index, Δ the quantization step and $\hat{X}$ the reconstructed value. In the case of the DC band, the reconstructed value $\hat{X}$ is computed as

$$\hat{X} = \begin{cases} Y & \text{if } d\Delta \le Y \le (d+1)\Delta \\ d\Delta & \text{if } Y < d\Delta \\ (d+1)\Delta & \text{if } Y > (d+1)\Delta \end{cases} . \tag{2.24}$$

For the AC bands, the reconstructed value $\hat{X}$ is computed in a similar way.

2.5 Performance Evaluation of DVC

The performance of the DVC codec [17] with respect to AVC/H.264 is discussed in this section for error-free and error-prone environments. For AVC/H.264, three modes are considered:

- AVC/H.264 Intra - encoding with AVC/H.264 without exploiting temporal redundancy. Each frame is independently encoded by exploiting the spatial redundancy.

- AVC/H.264 Inter - encoding with AVC/H.264 in IPP mode with a GOP size of 15.
- AVC/H.264 Inter No Motion - encoding with AVC/H.264 in IPP mode but without performing any motion estimation, which is the most computationally expensive encoding task. The GOP size is set to 15.

The Flexible Macroblock Ordering (FMO) feature is set to the dispersed mode at the encoder to improve the error resilience of AVC/H.264 [45]. This mode allocates the macroblocks to the different slices in a checkerboard-like pattern for better error resiliency as the different slices are independently encoded. This allows also for better error concealment as the samples of a missing slice can be efficiently estimated from neighboring samples belonging to correctly decoded slices.

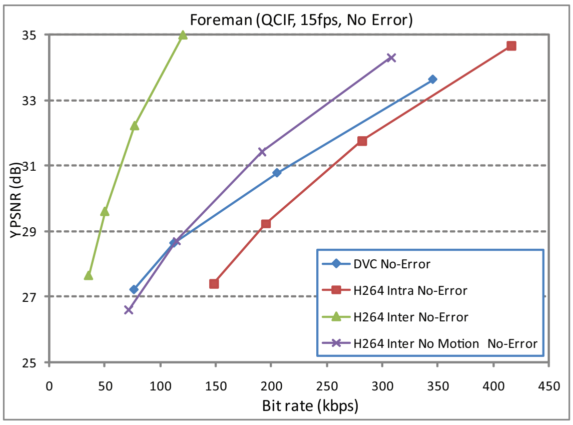
In this evaluation, only luminance data is encoded and a communication channel, characterized by the error pattern files provided in [46] with different Packet Loss Rates (PLRs). Test sequences *Foreman* and *Hallmonitor* in the QCIF format at 15 fps are corrupted with PLR values of 5%, 10% and 20%. In the error-prone case, the results are obtained by averaging over ten runs. Moreover, if the bit error probability of the decoded bit plane is higher than 10^{-3}, the decoder uses the corresponding bit plane from the SI. The header of the WZ bitstream, which contains critical information such as frame size, quantization parameters and Intra period is assumed to be correctly received.

Figure 2.20 (a) shows the RD performance for *Foreman* in an error-free environment. DVC is similar to AVC/H.264 No Motion at low bit rates. As the rate increase, the performance gap increases in favor of AVC/H.264 No Motion to reach a maximum gap of 1.5 dB. Moreover, DVC outperforms AVC/H.264 Intra at all bit rates. *Foreman* is a challenging video in terms of SI generation as it contains high activity in addition to camera pan. Nevertheless, DVC outperforms AVC/H.264 Intra, which is the closest codec to DVC in terms of encoding complexity.

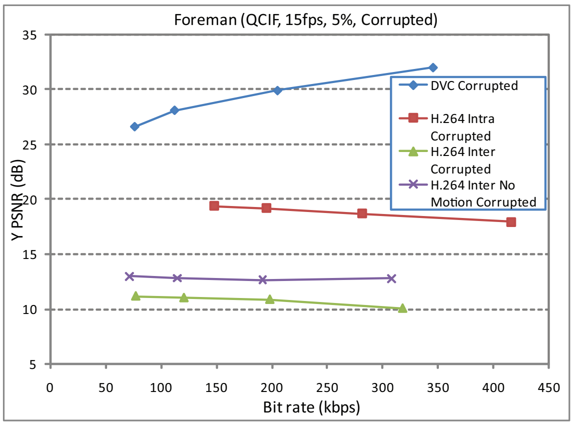
When the data is corrupted with packet losses (Figure 2.20 (b) and Figure 2.21 (a), (b)), DVC shows much more resistance to errors than the deterministic AVC/H.264 codec as DVC is based on a statistical approach. For the different AVC/H.264 modes, the Intra mode performs better than the other modes since each frame is independently encoded in the Intra mode. On the other hand, the other modes imply inter-frame prediction and by this favoring error propagation called the drift effect.

The RD performance for *Hallmonitor* is depicted in Figure 2.22(a) in the error-free case. As this sequence is very easy in terms of temporal prediction, DVC has a good RD performance, significantly outperforming AVC/H.264 Intra at all bit rates. Moreover, AVC/H.264 No Motion and Inter have a close RD performance outperforming DVC by around 4.0 dB. In the error-prone case (Figure 2.22 (a) and Figure 2.23 (a), (b)), Similar results to the *Foreman* case are obtained as DVC significantly outperforming the different modes of AVC/H.264. Thus, this strengthens the fact that DVC has good error resilience properties.

Finally, Figure 4.5 (b) shows the artifacts caused by packets lost for a conventionally coded stream with AVC/H.264 Intra. The lost data is reflected into really strong blocky artifacts as complete blocks are lost. For WZ data, the packets lost tend to cause artifacts around the edges as illustrated in Figure 4.5 (c).

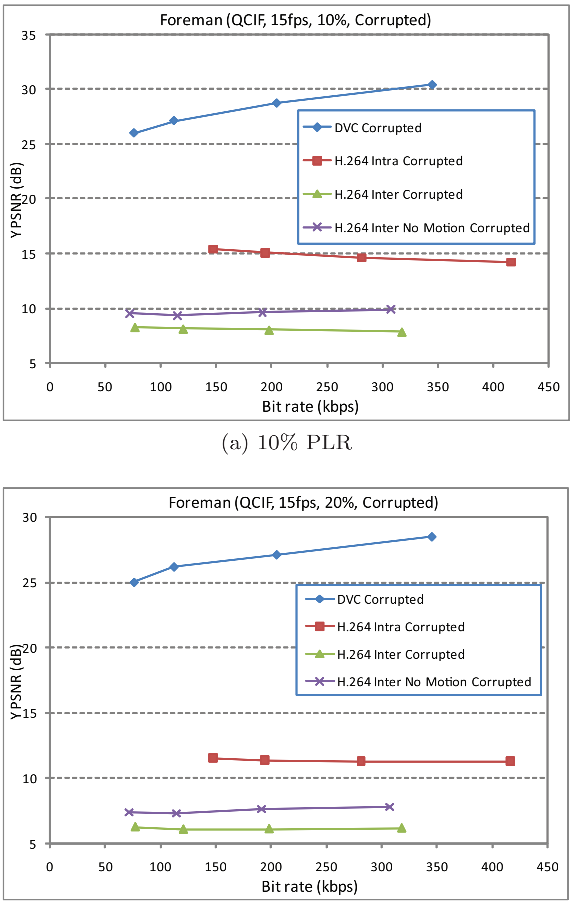


(a) No error



(b) 5% PLR

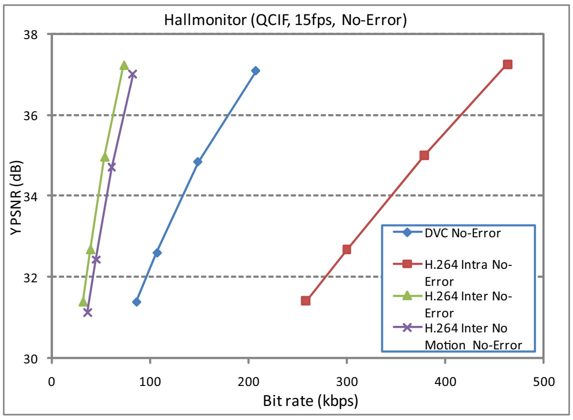
Figure 2.20: Performance evaluation for *Foreman*, DVC vs. AVC/H.264.



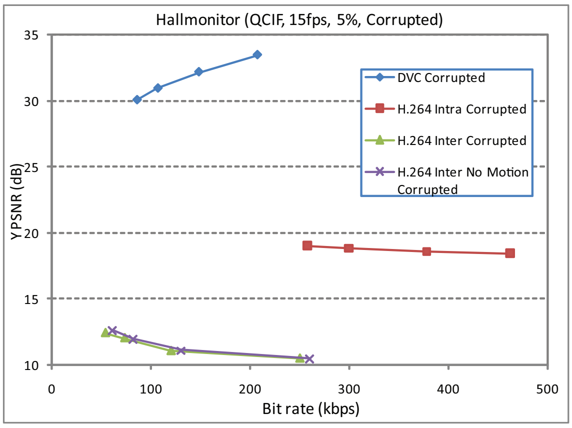
(a) 10% PLR

(b) 20% PLR

Figure 2.21: Performance evaluation for *Foreman*, , DVC vs. AVC/H.264.

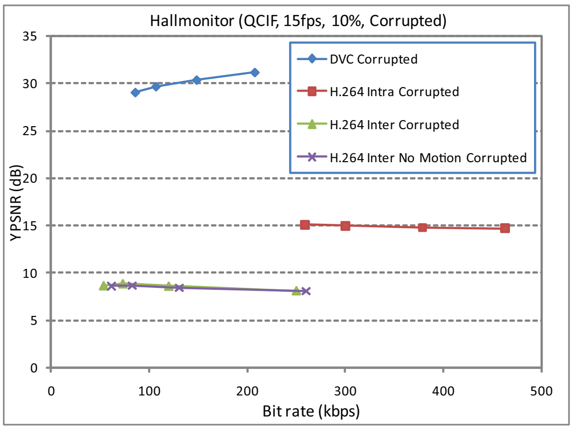


(a) No error

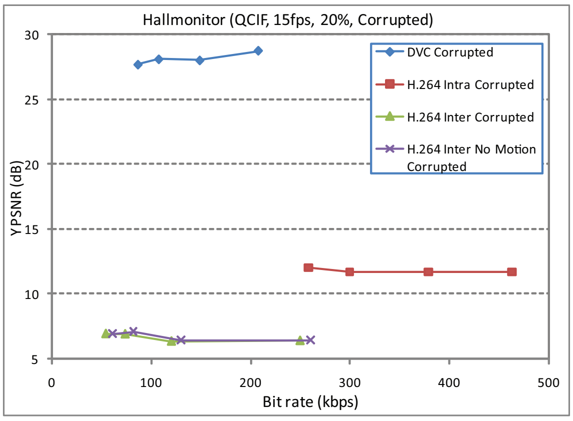


(b) 5% PLR

Figure 2.22: Performance evaluation for *Hallmonitor*, DVC vs. AVC/H.264.

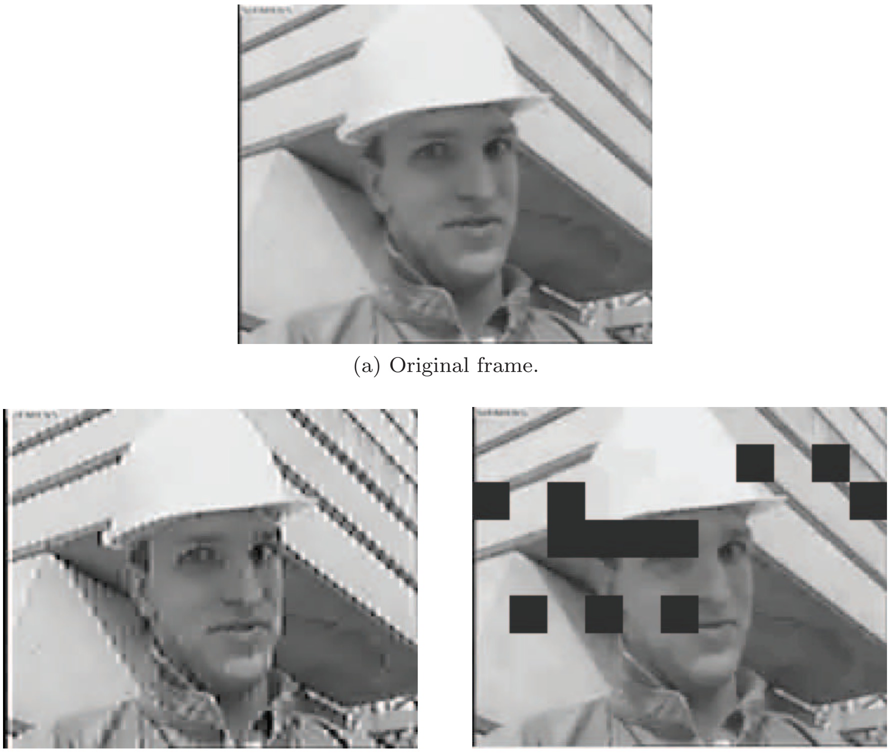


(a) 10% PLR



(b) 20% PLR

Figure 2.23: Performance evaluation for *Hallmonitor*, DVC vs. AVC/H.264.



(a) Original frame.

(b) Errors in the WZ data.

(c) Errors in AVC/H.264 Intra data.

Figure 2.24: Artifacts due to lost packets in AVC Intra and WZ data for *Foreman*.

2.6 Applications for DVC

Some promising DVC applications [47, 48] are identified in this section with a focus on application scenarios for which DVC may bring major advantages, highlighting also the drawbacks. More specifically, DVC strengths such as error resilience, encoder-decoder complexity tradeoff, low power encoder consumption are studied for each scenario according to its specific requirements.

2.6.1 DVC Functional Benefits

The analysis of the DVC and its statistical framework concludes that DVC offers the following benefits

- **Flexible encoder-decoder complexity allocation** - DVC provides the benefit of a flexible allocation of the complexity between the encoder and the decoder. DVC reduces the the encoding complexity when compared to convectional coding by shifting a part of it towards the decoder. Further, a particular case of this flexible allocation is the interesting case of low complexity encoding and decoding via transcoding. The latter would transform the compressed stream to a standard one (e.g. AVC/H.264), which enables low complexity decoding at the other end.

- **Improved error resilience** - Since DVC relies on a statistical framework, errors due to channel corruption are mitigated in time. The WZ bits do not only improve the quality of the SI but also recover from transmission errors.

- **Codec-independent scalability** - Most scalable codecs are based on a predictive approach from lower layers to upper layers. This requires the encoder's deterministic knowledge of previous layers in order to create the successive enhancement layers. On the other hand, DVC uses a correlation model and thus does not require the knowledge of the previous layers. This implies that lower layers may be generated by different codecs. This is called codec-independent scalability.

- **Separate encoding for multiview** - There are functional benefits of using DVC in a multiview video scenario as inter and intra-view correlations are exploited at the decoder side only. In this case, the DVC approach provides a significant architectural benefit since it implies independent encoding of the views. On the other hand, a typical predictive approach would exploit the inter-view correlation at the joint encoder requiring the various cameras to communicate among them, which is time consuming and requires complex networking.

2.6.2 DVC Application Scenarios

Wireless Low Power Surveillance

Surveillance is the process of monitoring the behavior of people, objects or processes within systems for conformity to expected or desired norms in trusted systems for security or social control [49]. Surveillance technology grows rapidly as security is becoming very important in our daily life. Homeland security is driving most of the new developments in this area but wireless video networks can play an important role as well in domestic applications such as baby surveillance (Figure 2.25).

Figure 2.25: Wireless camera for domestic surveillance equipped with a video receiver [50].

Wireless Surveillance can be used to monitor private and public spaces. It is also useful for other purposes such as military reconnaissance to gather information about the enemy. A camera video transmitter disguised as pens or other small object with spy or military purposes is another application that falls into this field as well (Figure 2.26).

Figure 2.26: Tiny surveillance camera [51].

Wild life monitoring [52] is another target application of DVC, where wireless cameras, placed in strategic points (Figure 2.27) like forests, national parks and natural reserves, monitor and alert authorities about disasters such as fires and avalanches. The system can be designed with transcoders to enable the usage of mobile devices at the reception of the videos and the alerts. Forest guards will be provided with such devices and can react immediately when the disaster is detected.

DVC has the advantage of resulting in low complexity encoders, which helps in reducing complexity and power consumption of such wireless devices. Moreover, since the data is transmitted in a wireless way, which is error-prone, DVC provides good error resiliency with respect to transmission errors. Further, DVC allows to encode the multiple camera streams independently. On the other hand, DVC has the drawback of having a poor RD performance for video content with

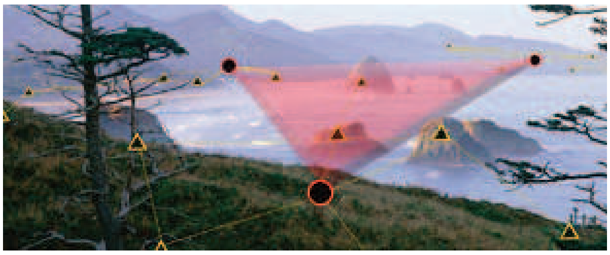
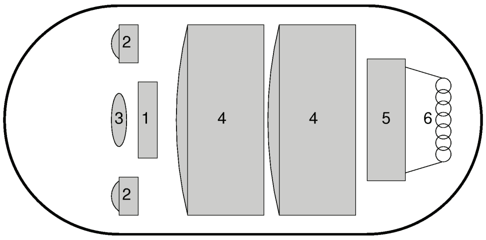


Figure 2.27: Monitoring wildlife is an emergent application for wireless low power surveillance networks [52].



(a) 1 - CMOS visual sensor, 2 - LEDs, 3 - lens, 4 - batteries, 5 - transmitter, 6 antenna.

(b) capsule endoscope called PillCam.

Figure 2.28: Wireless capsule endoscope [53].

high activity compared to conventional encoders. Finally, if the video is to be decoded by a low complexity device, this would require transcoding somewhere in the middle between the encoder and the decoder.

Wireless Capsule Endoscopy

Many diseases of the human body can only be spotted with images of the ill region. With X-ray, the whole body can be photographed but these images are not very accurate and not all diseases can be detected by this technique. X-ray studies may be unable to pinpoint exact locations of abnormalities. An example is to determine the source of gastrointestinal bleeding. Therefore, the capsule endoscopy [53] can be used for this purpose.

Capsule endoscopy is considered as a breakthrough in gastrointestinal diagnostics by offering several advantages over traditional radiological imaging such as high quality color images and remote consulting capabilities. The capsule has the size of a large pill and contains a battery, a strong light source, a camera and a small transmitter as shown in Figure 2.28.

In the context of wireless endoscopy, DVC offers the possibility of low complexity encoding and helps in maintaining the capsule simple, small and economic in terms of battery consumption. On the other hand, DVC is very high in terms of decoding complexity and prevents the display of video in realtime, which delays the diagnosis.

Video Conferencing with Mobile Phones

Video conferencing is the transmission of synchronized video and audio back and forth between two or more physically separate locations. It allows people at different locations to communicate with the feeling as if the participants are in the same physical place.

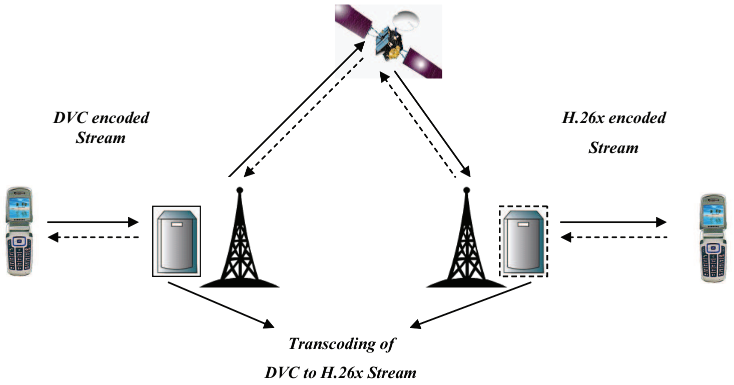


Figure 2.29: The transcoding process of WZ bit stream to a conventional H.26x bit stream [48].

In the case of mobile phones video conferencing, both end devices have limited computational resources and power. Therefore, the DVC bit stream should be transcoded to a standard one (e.g. AVC/H.264) inside the network thus enabling a low complexity encoding and decoding as depicted in Figure 2.29.
The main advantage of using DVC is enabling low complexity encoding or increasing the video resolution while power consumption and computational complexity are maintained. Further, DVC improves the error resilience properties of the mobile devices and should achieve a good compression efficiency when compared to conventional codecs. The reason for the latter is the low activity in video conferencing content, which produces a highly correlated SI with the WZ frames. Nevertheless, the bottleneck of the system is that the DVC stream needs to be decoded anyway for transcoding, which is not realtime yet and compromises the performance of the whole video conferencing system.

2.7 Conclusion

The theoretical foundations of DVC are first reviewed in this chapter and the different entities of the DVC codec used in this research are described. Then, it is compared against AVC/H.264 in error-free and error-prone environments. The simulation results show that DVC can be very competitive with AVC/H.264 in the Intra mode but it is inferior to the Inter mode. On the other hand, DVC exhibits good error resilience properties as it significantly outperforms AVC/H.264 in its different coding modes in error-prone environments. Finally, some potential applications where using DVC is beneficial are described. These applications are mainly characterized by the need for low complexity encoding and transmission over error-prone environments. DVC offers also additional benefits like codec-independent scalability and independent coding of cameras in a multiview scenario.

Chapter 3

Intra Frame Encoding

3.1 Introduction

Intra frame encoding is used in DVC to generate the conventional part of the compressed stream. The key frames are encoded using conventional Intra frame codecs such as JPEG [11], JPEG2000 [12], AVC/H.264 Intra [13] or JPEG XR [14], which is based on recent codec from Microsoft, HD Photo [54]. This chapter presents a review of these codecs and their evaluation for a rich set of video sequences with experimentally fine-tuned coding parameters for each codec.

JPEG [11] (Joint Photographic Experts Group) is a widely used method for image compression and the most common image format used by digital cameras and other photographic capture devices. Further, it is the most common format for storing and transmitting photographic images on the World Wide Web (WWW). It is based on the Discrete Cosine Transform (DCT) and achieves good compression ratios sufficient for a wide range of applications.

JPEG2000 [12] is a wavelet-based compression standard for still images. It was created by the JPEG committee [55]. Besides offering a number of new functionalities such as Region Of Interest (ROI) definition and scalability, it outperforms the original DCT-based JPEG standard in terms of compression efficiency.

AVC/H.264 [13], for Advanced Video Coding, is a digital video codec standard noted for achieving very high data compression. It is developed by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC Moving Picture Experts Group (MPEG) [1] as the product of a collective partnership effort known as the Joint Video Team (JVT) [56]. It is based on a block-based integer DCT-like transform and provides good video quality at substantially low bit rates. In addition, it performs spatial prediction for Intra frame encoding and temporal motion estimation for Inter frame encoding as it exploits both spatial and temporal correlation for a better compression efficiency. In this evaluation, both AVC Intra Main Profile (MP) and High Profile (HP) are considered.

JPEG XR [14] is a new still image compression algorithm from the JPEG committee for digital imaging applications. Its design aims at optimizing image quality and compression efficiency while enabling low complexity encoding and decoding at the same time. It uses some of the same fundamental building blocks as traditional image and video codecs, e.g. color conversion, transform quantization, coefficient scanning and entropy coding. The main novelty in JPEG XR lies in the design of these functional blocks. In particular, the algorithm uses the reversible Lapped Bi-orthogonal Transform (LBT) and an advanced coefficient coding.

A performance evaluation of AVC MP Intra and JPEG2000 is conducted in [57]. It is reported that AVC Intra performs better than JPEG2000 in terms of RD performance for low and intermediate spatial resolution sequences. The gain of AVC Intra over JPEG2000 in PSNR is around 0.5∼2.0 dB. On the other hand, JPEG2000 performed better for higher resolution sequences with a gain around 0.5∼1.0 dB in PSNR. Furthermore, AVC HP Intra and JPEG2000 are compared in [58] for monochromatic still image encoding. It is shown that their performances are identical. Nevertheless, JPEG2000 has a gain of 1.0 dB in PSNR over AVC HP Intra if the 8x8 transform is disabled for the encoder. However, the evaluation is performed on a small set of images, which reduces its consistency. Finally, the same comparison is performed in [58] and [59, 60] but using high resolution video instead of still images. The experimental results in [59, 60] show that AVC HP Intra offers a RD gain around 0.2∼1.0 dB in PSNR over JPEG2000.

In [61], JPEG2000 is compared to both AVC Intra profiles considering a set of sequences with various spatial resolutions. It is shown that JPEG2000 is very competitive with AVC HP Intra with around 0.1 dB difference in PSNR in favor of AVC HP Intra for high spatial resolution sequences. On the other hand, JPEG2000 outperforms AVC MP Intra with gains around 0.1∼1.0 dB in PSNR. For intermediate and low spatial resolution sequences, both profiles of AVC Intra outperform JPEG2000. Moreover, [62] compares the performance of JPEG2000 and AVC HP Intra considering a set of high definition video sequences and better tuned compression parameters when compared to [61]. Results show quite competitive performance between both coding approaches, while in some cases, AVC HP Intra outperforms JPEG2000. Finally, the RD performance of AVC HP Intra, JPEG2000 and JPEG XR for a number of different high resolution still images is presented in [63]. The results show that JPEG2000 clearly outperforms both AVC HP Intra and JPEG XR with gains around 0.5∼2.0 dB in PSNR, considering the luma component of the image in the YUV 4:4:4 space. Furthermore, AVC HP Intra is very competitive with JPEG XR with around 0.2∼0.8 dB difference in PSNR in favor of AVC HP in most cases.

A RD performance evaluation of the codecs described above using the PSNR distortion metric is described in this chapter. A rich set of test sequences with different spatial resolutions ranging from QCIF (176x144) up to 1080p (1920x1080) is used in this evaluation. Moreover, the different compression parameters are experimentally fine-tuned to obtain the best compression efficiency for each codec, which is not the case of evaluations described in the literature.

The remaining of this chapter is structured as follows. The evaluated codecs, JPEG, JPEG2000, AVC/H.264 Intra and JPEG XR, are reviewed in sections 3.2, 3.3, 3.4 and 3.5, respectively. The test material and the evaluation methodology are discussed in section 3.6 in addition to the RD comparison between the different codecs. Finally, the conclusion regarding the evaluation is presented in section 3.7.

3.2 JPEG

JPEG stands for Joint Photographic Experts Group, which is the name of the committee that created the standard. The JPEG lossy compression algorithm is depicted in Figure 3.1.

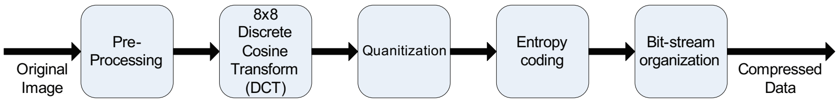


Figure 3.1: The JPEG fundamental building blocks.

In the pre-processing stage, the samples are transformed from the RGB color space to YC_bC_r. The chroma components are downsampled as the human eye sees considerably more fine details in the brightness of an image (Luminance) than in the color of an image (Chroma). Therefore, downsampling the chroma component improves the compression efficiency without drastically affecting the perceptual quality. Then, the samples are shifted from an unsigned integer range to a signed integer interval centered around 0. Further, they are transformed using a block-based Discrete Cosine Transform (DCT) [64] to generate 64 coefficients, one DC coefficient and 63 AC coefficients, per each 8x8 block. The DCT concentrates the energy of the input signal into the lower frequencies as most of the high frequency coefficients are zero or near zero. The DCT coefficients are then quantized using a matrix defining a quantization step per coefficient. The quantization coefficient, $F_q(u, v)$, is computed as the nearest integer of the division of the frequency coefficient, $F(u, v)$, and the corresponding quantization step, $q(u, v)$.

$$F_q(u,v) = Round(\frac{F(u,v)}{q(u,v)}). \tag{3.1}$$

The Dequantization to recover $\hat{F_q}(u, v)$ is done by multiplying the quantized frequency by the corresponding quantization step.

$$\hat{F_q}(u,v) = F_q(u,v)q(u,v). \tag{3.2}$$

Because of the high correlation between the DC coefficients of adjacent blocks, the DC coefficient is coded predictively using the previous block's DC coefficient as a reference. The rest of the coefficients are organized in a zig-zag order by placing the low frequency coefficients before the high ones as shown in Figure 3.2. Finally, the quantized coefficients are entropy coded [65, 66] and organized to from the compressed bit stream.
For more details on JPEG refer to [11].

3.3 JPEG2000

The JPEG2000 [12] standard makes use of the Discrete Wavelet Transform (DWT) [67]. JPEG2000 supports some important features such as improved compression efficiency over JPEG, lossless and lossy compression, multi-resolution representation, Region Of Interest (ROI) coding, error resilience and a flexible file format. Figure 3.3 depicts the JPEG2000 fundamental building blocks.

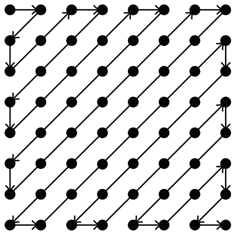


Figure 3.2: Zig-zag order of the DCT coefficients.

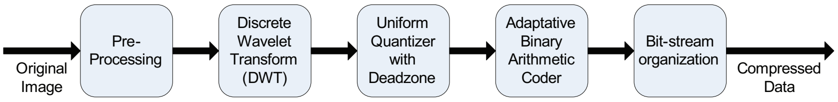

Original Image
Pre-Processing
Discrete Wavelet Transform (DWT)
Uniform Quantizer with Deadzone
Adaptative Binary Arithmetic Coder
Bit-stream organization
Compressed Data

Figure 3.3: The JPEG2000 fundamental building blocks.

In the Pre-Procssing stage, an inter-component transformation is used to decorrelate the color data. There are two possible transforms. The first transform is the Irreversible Color Transform (ICT), which is identical to the traditional RGB to YC_bC_r color transformation and can only be used for lossy encoding.

$$Y = 0.229(R - G) + G + 0.114(B - G). \tag{3.3}$$

$$C_b = 0.564(B - Y), C_r = 0.713(R - Y).$$

The other transform is the Reversible Color Transform (RCT), which is a reversible integer to integer transform that approximates the ICT for color decorrelation and can be used for both lossless and lossy encoding.

$$Y = \lfloor R + 2G + \frac{B}{4} \rfloor. \tag{3.4}$$

$$U = R - G, V = B - G.$$

The inverse RCT which is capable of exactly recovering the original RGB signal is defined as follows

$$G = Y - \lfloor U + \frac{V}{4} \rfloor. \tag{3.5}$$

$$R = U + G, B = V + G.$$

Then, the DWT is applied to the processed samples. A one dimensional (1D) DWT at the encoder can be seen as the application of a low pass and a high pass filter, followed by a downsampling after each filtering operation as shown in Figure 3.4.

The pair of filters (h_0, h_1) is known as the analysis filter bank [67]. The low pass filter preserves low frequencies of the signal while attenuating or eliminating the high frequencies, thus resulting in a blurred version of the original signal. On the other hand, the high pass filter preserves the high frequencies in the signal such as edges, texture and details, while removing or attenuating the low frequencies. The pair of filters (h_0, h_1) is designed in such a way that after downsampling the output of each filter, the original signal can be completely recovered from these samples. This is referred to as the perfect reconstruction

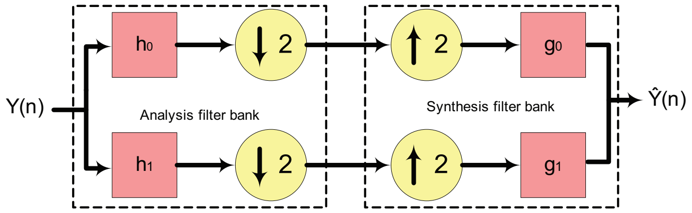


Figure 3.4: 1D wavelet analysis and synthesis filter bank.

property. This is performed using the pair of filters (g_0, g_1), known as the synthesis filter bank [67]. The samples are upsampled by inserting zeros between every two samples and then filtered by g_0 and g_1. Both results are summed up to produce the reconstructed signal. For perfect reconstruction, the analysis and synthesis filters should satisfy the following two conditions

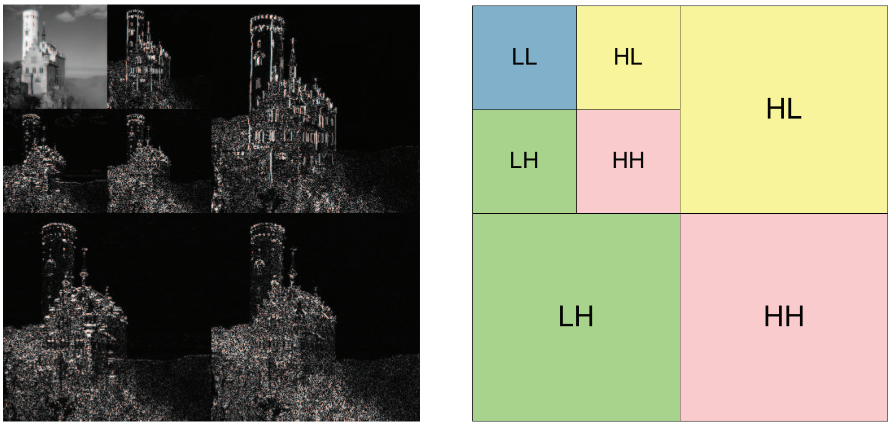
$$H_0(z)G_0(z) + H_1(z)G_1(z) = 2, \tag{3.6}$$
$$H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0,$$

where $H_{0,1}(z)$ is the Z-transform of $h_{0,1}$ and $G_{0,1}(z)$ is the Z-transform of $g_{0,1}$. These conditions can be satisfied by choosing $G_0(z) = cz^{-l}H_1(-z)$ and $G_1(z) = cz^{-l}H_0(-z)$ where l is an integer constant and c is a scaling factor. The filter bank satisfying these conditions is known as the bi-orthogonal filter bank [67]. This comes from the fact the h_0 and g_1 are orthogonal to each other and the same applies to h_1 and g_0. Furthermore, the analysis filters have to be of unequal lengths to satisfy these conditions.

The 1D DWT can be easily extended to 2D by applying the filter bank in a separable way. At each level of the DWT decomposition, each row of the image is first transformed using the 1D analysis filter bank. Then, the same filter is applied vertically to each column of the filtered and downsampled data. The result of one wavelet decomposition is 4 filtered and downsampled images, called subbands. The lowest frequency subband (denoted as the LL band to indicate low pass filtering in both directions) can be further decomposed into 4 smaller subbands as shown in Figure 3.5 (b). This process may be repeated until no tangible gains in compression efficiency can be achieved. The DWT provides also a multi-resolution image representation as shown in Figure 3.5 (a). The DWT enables very good compression due to its good energy compaction and ability to decorrelate the image across a larger scale.

The resulting wavelet coefficients are quantized using a uniform quantizer with a central deadzone. It is shown that this quantizer is optimal for a continuous signal with a Laplacian distribution such as DCT or wavelet coefficients [12]. For each subband b, a basic quantization step size Δ_b is used to quantize all the coefficients in this subband. The choice of Δ_b can be driven by the perceptual importance of each subband based on the Human Visual System (HVS) or other considerations such as rate control. The quantization maps a wavelet coefficient $y_b(u,v)$ in subband b to a quantized coefficient $q_b(u,v)$ according to the following rule

$$q_b(u,v) = sign(y_b(u,v))\lfloor\frac{|y_b(u,v)|}{\Delta_b}\rfloor. \tag{3.7}$$



(a) The resulting subbands.

(b) The filtering applied to obtain the corresponding subband.

Figure 3.5: Illustration of the DWT with 2 decomposition levels. L means low pass filtering in a given direction, horizontal or vertical. H means high pass filtering in a given direction, horizontal or vertical.

When the synthesis filter bank is used, the reconstructed transform coefficient, $\hat{y_q}(u,v)$ is computed as follows

$$\hat{y_q}(u,v) = \begin{cases} (q_b(u,v)+\alpha)\Delta_b & \text{if } q_b(u,v) > 0 \\ (q_b(u,v)-\alpha)\Delta_b & \text{if } q_b(u,v) < 0 \\ 0 & \text{Otherwise} \end{cases}$$

, where $0 \leq \alpha < 1$ is an arbitrary parameter chosen by the decoder.
The quantized coefficients are gathered in subbands. Each subband is partitioned into small rectangular blocks called codeblocks where each codeblock is independently coded by an adaptive binary arithmetic coder [68]. Finally, the output of the arithmetic coder is organized as a compressed bitstream which offers a significant degree of flexibility. This enables features such as random access, region of interest coding and scalability. This flexibility is achieved partly through the various structures of components, tiles, subbands, resolution levels, and codeblocks.
For more details on JPEG2000 refer to [12].

3.4 Advanced Video Coding (AVC)/H.264 Intra

AVC MP Intra is based on the block-based integer DCT-like transform. The different stages for encoding an input image are shown in Figure 3.6.

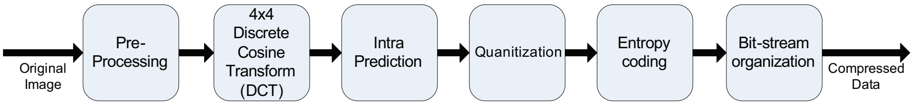


Figure 3.6: AVC Intra encoder.

Unlike its predecessors, the block size for the transform is reduced from 8x8 to

4x4. Furthermore, the 4x4 transform has the advantage of being simple and its core is multiply free as it only requires additions and shifts. AVC Intra takes advantage of the spatial correlation to improve the coding efficiency. The Intra coding of a macroblock consists in four main steps, spatial prediction, 4x4 DCT transform, scalar quantization and entropy coding. If a macroblock is encoded in Intra predictive mode, a prediction macroblock is formed based on previously encoded and decoded macroblocks within the same frame. This prediction macroblock is subtracted from the current macroblock prior to encoding. For the luminance samples, the prediction macroblock is computed for each 4x4 subblock or for a 16x16 macroblock. There are a total of 9 optional prediction modes for each 4x4 luminance subblock, 4 optional modes for a 16x16 luminance macroblock and 4 modes for each 8x8 chrominance macroblock.

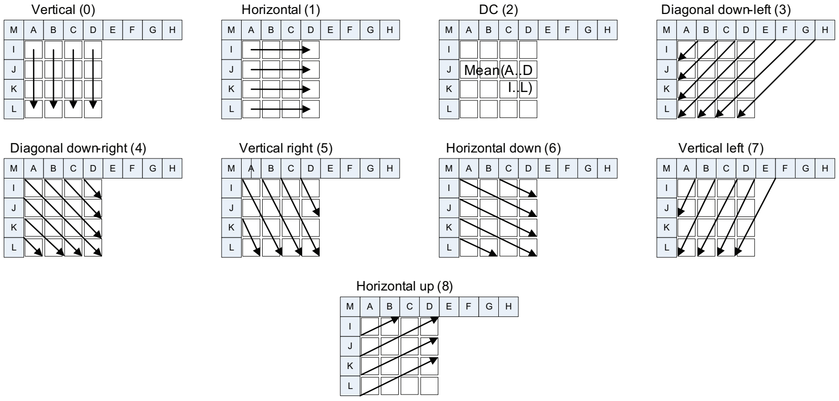


Figure 3.7: 4x4 luminance prediction modes.

The arrows in Figure 3.7 indicate the prediction direction in each mode. For modes 3 to 8, the predicted samples are formed from a weighted average of the prediction samples A-M. For mode 2, the predicted samples are formed from the mean of samples A to D and I to L. For modes 0 and 1, the samples are predicted vertically and horizontally from samples A to D and I to L respectively. The encoder selects the prediction mode for each block that minimizes the prediction residual.

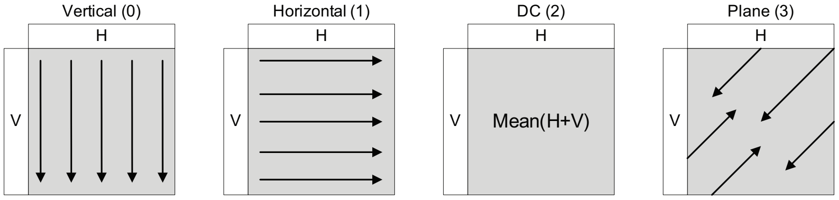


Figure 3.8: 16x16 luminance prediction modes.

Figure 3.8 shows the 4 modes for the 16x16 luminance macroblocks. In mode 3, a linear plane function is fitted to the upper and left hand samples H and V. The 4 chroma prediction modes are very similar to the 16x16 luma prediction modes, except that the order of mode numbers is different: DC (mode 0), hori-

zontal (mode 1), vertical (mode 2) and plane (mode 3).
Then, the macroblocks are transformed using a 4x4 separable integer transform with properties similar to that of a DCT. Furthermore, a non-uniform scalar quantization is used to quantize the transformed coefficients where each macroblock has a Quatization Parameter (QP). Finally, the quantized coefficients are entropy coded. AVC Intra supports two modes for entropy coding, Context-Adaptive Variable Length Coding (CAVLC) [69] and Context-Adaptive Binary Arithmetic Coding (CABAC) [70]. With CAVLC, multiple VLC tables are available and the encoder switches among them based on previously encoded syntax elements. On the other hand, CABAC is based on an arithmetic coder. Using arithmetic coding, each symbol of the alphabet can be assigned a non-integer number of bits, therefore outperforming VLC tables. Furthermore, a context model is built based on the statistics of previously transmitted syntax elements in order to better estimate conditional probabilities. This allows for the adaptation to non-stationary statistics. For these reasons, CABAC achieves substantially better coding performance when compared to CAVLC. However, it requires significantly higher computational complexity.
Further, the AVC MP is extended by introducing the High Profile encoder or AVC Fidelity Range Extensions (FR Ext) [71]. In the latter, an 8x8 integer transform is introduced. The encoder chooses adaptively between the 4x4 and the 8x8 transform for the luminance samples. Using the 8x8 transform enables the preservation of fine details and textures which generally require larger basis functions. Furthermore, three sets of context models are added in CABAC for the 8x8 transform coding. Meanwhile, CAVLC is used by splitting the 8x8 transform coefficients into groups of 4x4. Furthermore, AVC HP allows using more than 8 bits per sample for more accuracy. In addition, The High Profile supports higher color space resolutions such as YUV 4:2:2 and YUV 4:4:4 with interesting features such as scaling matrices for perceptually tuned and frequency-dependent quantization specified at the encoder, the reversible residual color from RGB to YCgCo transform which is applied only to residual data and the lossless coding capability.
For more details on AVC/H.264 please refer to [13, 71].

3.5 JPEG XR

The JPEG XR [14] encoding structure is composed of the following stages, color transform, reversible Lapped Bi-orthogonal Transform (LBT), and transform coefficient coding. The LBT is an integer hierarchical two stage lapped transform, used to convert spatial domain image data to frequency domain information. The transform is based on a flexible concatenation of two operators, the Photo Core Transform (PCT) and the Photo Overlap Transform (POT). The PCT is similar to the widely used DCT. It exploits spatial correlation within the block but has some of the same shortcomings of the DCT and other block-based transforms. It does not exploit redundancy across block boundaries and sometimes results in blocking artifacts at low bit rates. The POT is designed to exploit the correlation across block boundaries to improve the compression efficiency. It mitigates blocking artifacts as well, thus addressing the drawbacks of the PCT. It is similar to the Normative deblocking filter used in AVC/H.264. For more details on the PCT and the POT please refer to [14, 54, 72, 73].

The transform used in JPEG XR has a two stage hierarchical nature (Figure 3.9) defined as follows

- A 4x4 POT is applied if it is enabled. Each 4x4 block within a 16x16 macroblock undergoes a first stage PCT, yielding 1 DC coefficient and 15 AC coefficients.
- The first stage DC coefficients of all 4x4 blocks within a 16x16 macroblock (16 DC coefficient for one macroblock) are collected to form a single 4x4 DC block. Another POT is performed if it is enabled at the second stage. A second PCT stage is applied to the DC block. This yields 16 new coefficients, 1 second stage DC component and 15 second stage AC components for the DC block. These are referred to respectively as the DC and Low Pass (LP) coefficients of the original macroblock. The LP band of a macroblock is composed of all the AC coefficients of the second transform stage. The remaining 240 coefficients, i.e. AC coefficients of the first transform stage of the 16x16 macroblock, are referred to as the High Pass (HP) coefficients of the macroblock.

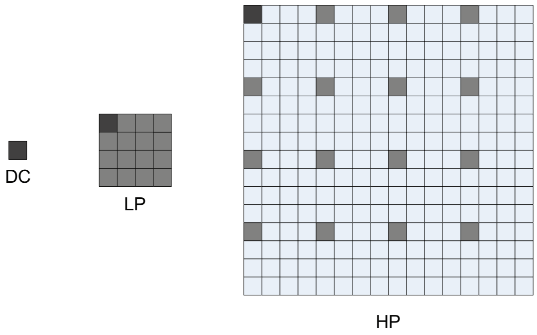


Figure 3.9: Macroblock frequency hierarchy.

At each stage, the PCT is always performed, while the overlap operator is functionally independent from the core transform, and can be switched on or off. There are three options for controlling the POT, disabled for both stages, enabled for first stage but disabled for second one and finally enabled for both stages. If the POT and the PCT are concatenated, the combined transform is equivalent to the LBT.

A 2x2 POT and 2x2 PCT are applied to YUV 4:2:2/ YUV 4:2:0 chroma channel. For the chroma channel of YUV 4:2:2, an additional 2-point Hadamard transform is applied to the DC coefficient of the second stage transform. The transform coefficients are grouped into three subbands with the same nomenclature, DC band, LP band, and HP band.

The key elements of the transform coefficient coding scheme is summarized in the following steps

- Independent coding of the transform bands: The DC, LP and HP bands are quantized and coded independently.
- Flexible coefficient quantization: The Quantization Parameters (QP) can be varied based on the location within the image, e.g. at frame, tile or

macroblock level, frequency band and color channel. The signaling scheme is designed to limit the overhead incurred by QP related side information.

- Inter block coefficient prediction: To remove inter block redundancy in the quantized transform coefficients while minimizing the storage requirements, there are three levels of inter block prediction, prediction of the DC subband coefficients, prediction of LP subband coefficients, and prediction of the HP subband coefficients. Inter block coefficient prediction is similar to the Intra prediction used in AVC/H.264.
  - The DC coefficient prediction has 4 modes, no prediction mode, prediction from the left neighbor, prediction from top neighbor and prediction from left and top neighbors. In the latter, the predictor is equal to the average neighbors.
  - The LP and HP coefficients prediction has 3 modes, no prediction mode, prediction from left and prediction from top.
- Layered coding of high frequency coefficients: The high frequency coefficients are partitioned into two components by adaptive coefficient normalization. After the partition, the significant information is entropy coded and the remainder is signaled using fixed length codes.
- Adaptive coefficient scanning: Scan arrays are used to convert the 2D transform coefficients within a block into a linear 1D encodable array. Scan patterns are adapted dynamically based on the local statistics of the coded coefficients. Coefficients with higher probability of non-zero values are scanned first.
- Advanced entropy coding using adaptive VLC table switching. A small set of fixed VLC tables are defined for each syntax element. The best table is then selected based on the local coefficient statistics. The choice of the table is made based on previously coded values of the corresponding syntax element. The adaptive context-based VLC and the separate entropy coding methods based on frequency division are similar to the choice of CAVLC and CABAC in AVC/H.264.

For more details on JPEG XR refer to [14].

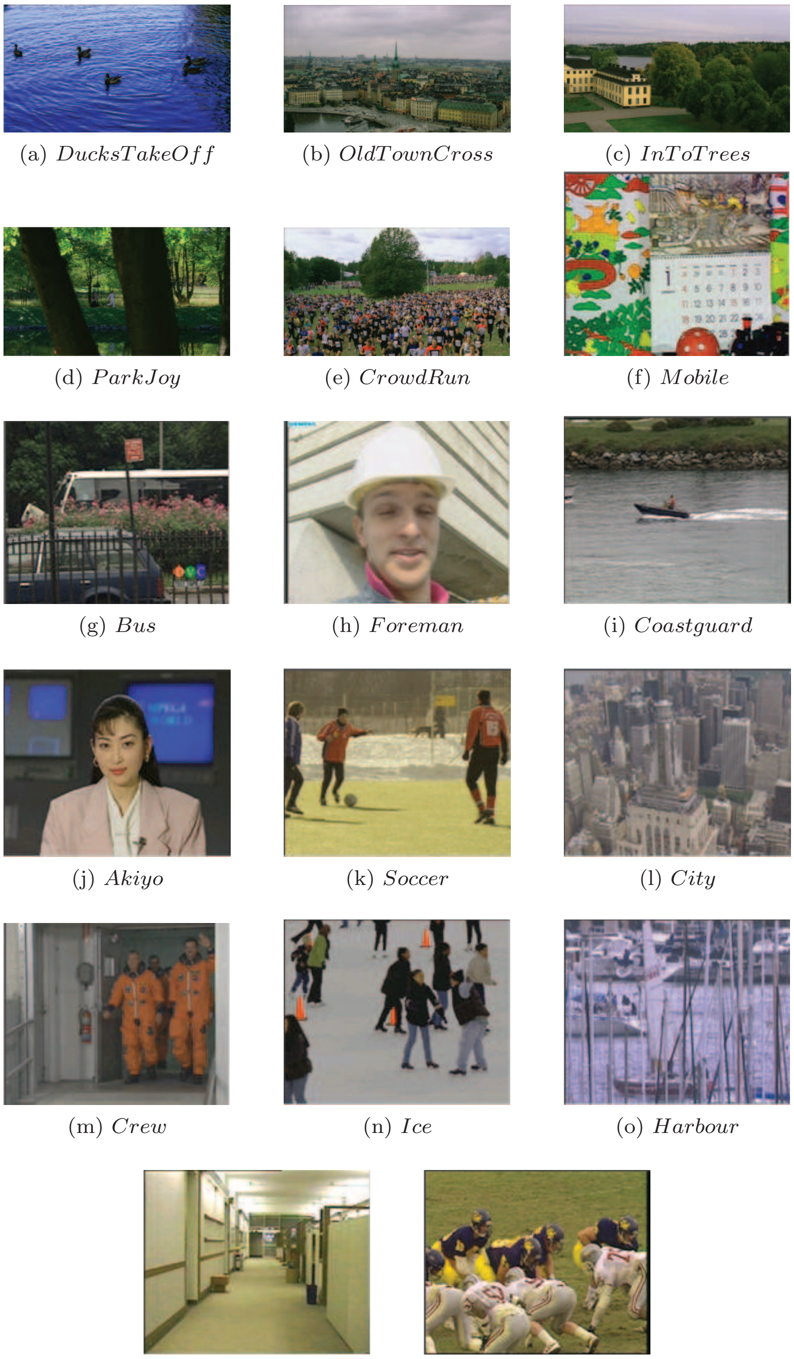
3.6 Simulation Results

3.6.1 Test Material and Evaluation Methodology

A rich set of videos is used in this evaluation as it contains 16 main sequences at different spatial resolutions. The set contains sequences with high texture such as *OldTownCross*, *City*, *Mobile* and *Bus*. On the other hand, *CrowdRun*, *Coastguard*, *Soccer*, *Harbour* and *Football* contain more or less uniform texture with significant motion and sometimes camera pan. Furthermore, *Crew* contains uniform texture with sudden illumination changes due to camera flashes. *Hallmonitor* is a surveillance-like video sequence. Finally, *Akiyo* and *Foreman* are typical video conferencing sequences. Table 3.1 depicts the sequences used, their frame rate and spatial resolution and Figure 3.10 depicts the first frame

Table 3.1: Evaluation video set.

Sequence	Fps	Resolution	Sequence	Fps	Resolution
DucksTakeOff	50	1280x720	*OldTownCross*	50	1280x720
	50	1920x1080		50	1920x1080
InToTree	50	1280x720	*ParkJoy*	50	1280x720
	50	1920x1080		50	1920x1080
CrowdRun	50	1280x720	*Mobile*	7.5	176x144
	50	1920x1080		30	352x288
Bus	7.5	176x144	*Foreman*	7.5	176x144
	30	352x288		30	352x288
Coastguard	15	176x144	Akiyo	30	176x144
	30	352x288		30	352x288
Soccer	15	176x144	*City*	15	176x144
	15	352x288		15	352x288
	30	704x576		30	704x576
Crew	15	176x144	*Ice*	15	176x144
	15	352x288		15	352x288
	30	704x576		30	704x576
Harbour	15	176x144	*Hallmonitor*	15	176x144
	15	352x288		30	352x288
	30	704x576	-	-	-
Football	7.5	176x144	-	-	-
	15	352x288	-	-	-



(a) *DucksTakeOff* (b) *OldTownCross* (c) *InToTrees*

(d) *ParkJoy* (e) *CrowdRun* (f) *Mobile*

(g) *Bus* (h) *Foreman* (i) *Coastguard*

(j) *Akiyo* (k) *Soccer* (l) *City*

(m) *Crew* (n) *Ice* (o) *Harbour*

(p) *Hallmonitor* (q) *Football*

Figure 3.10: The first frame of each video sequence.

from each test sequence.
Next, the different codec implementations used in this evaluation are presented with the corresponding compression parameters. These parameters have been chosen after experimentation in order to maximize the performance of each codec.
For JPEG, the Independent JPEG Group's free software [74] is used in the baseline mode.
The software KAKADU [75] version 4.4 is used for the JPEG2000 compression with the following settings

- One tile per frame.
- 3 decomposition levels for QCIF and CIF resolutions and 5 for the remaining spatial resolutions.
- Visual Frequency Weighting is switched-off. This parameter is used to give good visual appearance. On the other hand, it worsens the PSNR RD performance.
- Base step parameter (QStep) adapted per sequence and rate control switched-off. For QCIF and CIF resolutions, the performance is similar with and without rate control.

For the AVC/H.264 Intra coding, the publicly available reference software, JM version 11.0 [76], is used with the following settings

- Main and High Profile encoding.
- CABAC for High Profile.
- The 8x8 transform enabled for High Profile.
- Disable transform coefficients thresholding.
- Enable the use of explicit lambda parameters and set the weight of the Intra slice to 0.5.
- AdaptiveRounding is enabled. This parameter is used in the quantization process to adjust the rounding offset to maintain an equal expected value for the input and output of the quantization process for the absolute value of the quantized data. It is recommended to use AdaptiveRounding when encoding with high quality.
- AdaptRndPeriod is set to 1, AdaptRndWFactorIRef is set to 8 and AdaptRndWFactorINRef is set to 8. These parameters are associated with AdaptiveRounding.
- OffsetMatrixPresentFlag is disabled.
- Enable RD optimization.

The High Profile of AVC/H.264 is introduced to improve the compression efficiency for high spatial resolutions. Therefore, the Main Profile is evaluated only for small and intermediate spatial resolutions (QCIF, CIF and 704x576) and omitted for high spatial resolutions (720p and 1080p).

For JPEG XR, The HD Photo Device Porting Kit Specification 1.0 [77] is used with the default settings.
For the evaluation, the Peak To Noise Ratio (PSNR) is used as the distortion metric. The PSNR is computed as follows

$$MSE_k = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(I_k(n,i,j) - \hat{I}_k(n,i,j))^2,$$

$$PSNR_k = 20log_{10}(\frac{255}{\sqrt{MSE_k}}),$$

$$PSNR = \frac{1}{F}\sum_{k=1}^{F}PSNR_k,$$

where $\hat{I_k}$ is the k^{th} frame from the distorted video signal, I_k is the k^{th} frame from the original video signal, (N, M) is the spatial resolution of the video, F is the total number of frames and, MSE_k and $PSNR_k$ are respectively the Mean Square Error and Peak Signal to Noise Ratio for the k^{th} frame from the distorted video signal.

3.6.2 RD Performance

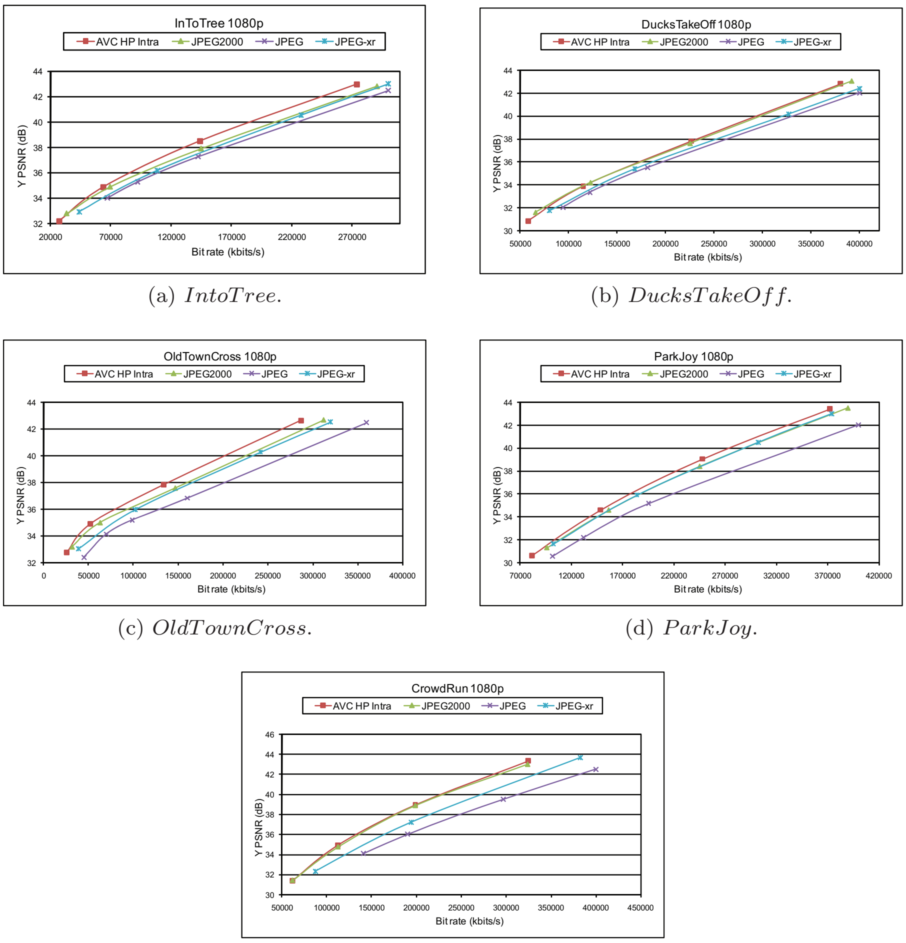
In this section, the RD performance of the different codecs for the set of video sequences is analyzed.

1080p spatial resolution

AVC HP Intra and JPEG2000 have a similar performance outperforming JPEG XR by around 1.0 and 2.0 dB for *DucksTakeOff* and *CrowdRun* respectively as shown in Figure 3.11. Furthermore, AVC HP Intra has a better performance than JPEG2000 especially at high bit rates by around 1.0 dB for *InToTree* and *OldTownCross*. Finally, AVC HP Intra outperforms JPEG2000 and JPEG XR for *ParkJoy* by around 0.5 dB. For all sequences, JPEG is outperformed by the other codecs with a maximum performance gap around 3.0 dB for *CrowdRun* with respect to AVC HP Intra. This gap is much smaller for *InToTree* and *DucksTakeOff* as it is around 1.0 dB.

720p spatial resolution

As depicted in Figure 3.12, AVC HP Intra and JPEG2000 have a similar performance with a slight advantage, around 0.1∼0.3 dB, in favor of AVC HP Intra for *CrowdRun* and *DucksTakeOff*. On the other hand, JPEG2000 outperforms AVC HP Intra at low and average bit rates by around 0.1∼0.5 dB for *DucksTakeOff*. Furthermore, both JPEG2000 and AVC HP Intra outperform JPEG XR by a maximum gap of 2.0 dB. For the sequences *OldTownCross*, *InToTree* and *ParkJoy*, clearly the performance gap tends to increase with bit rate in favor of AVC HP Intra with respect to JPEG2000. AVC HP Intra outperforms JPEG2000 for *OldTownCross* and *InToTree* by a maximum gap of 0.8 dB at high bit rates. This can be explained by the fact that both sequences contain significant areas with more or less uniform regions. This goes in favor of the



(a) *IntoTree*.

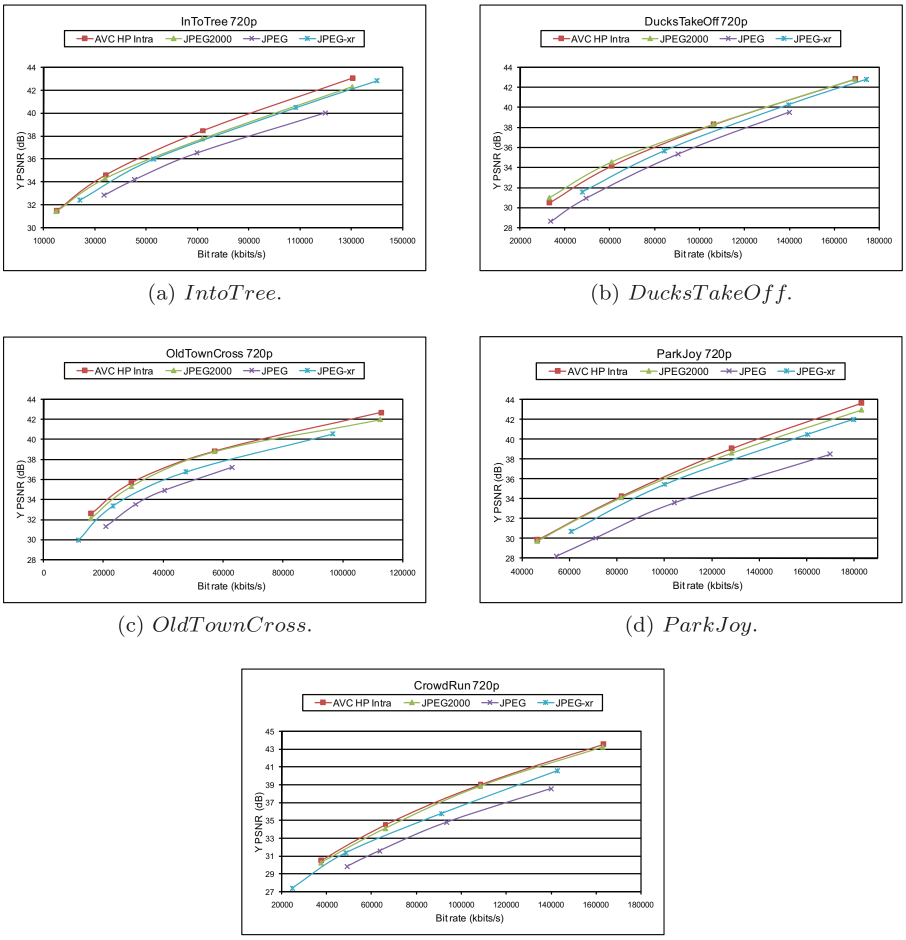
(b) *DucksTakeOff*.

(c) *OldTownCross*.

(d) *ParkJoy*.

(e) *CrowdRun*.

Figure 3.11: RD performance for 1080p spatial resolution.



(a) *IntoTree*.

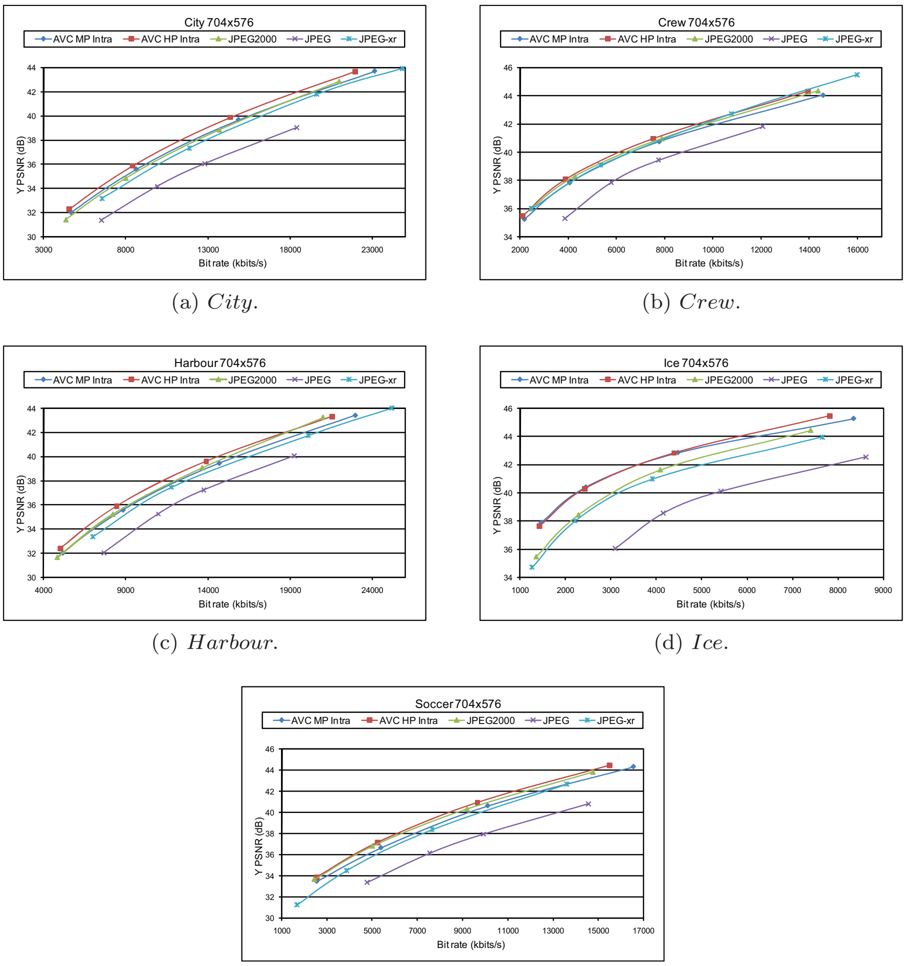
(b) *DucksTakeOff*.

(c) *OldTownCross*.

(d) *ParkJoy*.

(e) *CrowdRun*.

Figure 3.12: RD performance for 720p spatial resolution.



(a) *City*.

(b) *Crew*.

(c) *Harbour*.

(d) *Ice*.

(e) *Soccer*.

Figure 3.13: RD performance for 704x576 spatial resolution.

AVC Intra encoding since it takes better advantage of the spatial correlation. Finally, AVC HP Intra outperforms JPEG2000 for *ParkJoy* by around 0.1∼0.5 dB. On the other hand, JPEG2000 outperforms JPEG XR for *ParkJoy* and *OldTownCross* by around 0.7∼1.0 dB and they have a similar performance for *InToTree*.

704x576 spatial resolution

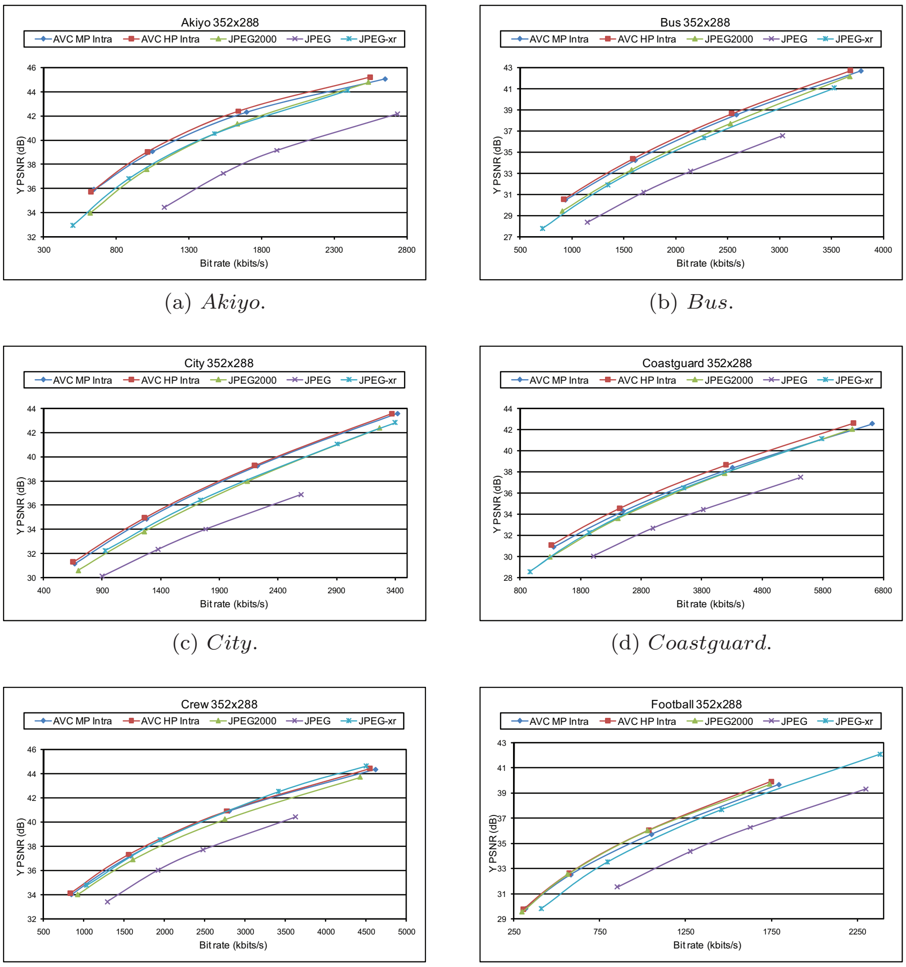
It is reported in [59] that for the *City* sequence, AVC HP Intra outperforms JPEG2000 with a gain around 1.0 dB in PSNR. This is not our case since the gain of AVC HP over JPEG2000 is around 0.6 dB at average bit rates and 0.3 dB at high bit rates as shown in Figure 3.13. In addition, it is reported in [57,59] that AVC HP Intra outperforms JPEG2000 for the *Harbour* sequence with a gain around 0.8∼1.0 dB. In our case, the gain is around 0.2 in favor of AVC HP Intra at average bit rates and in favor of JPEG2000 at high bit rates. Finally, it is reported in [57, 59] that AVC HP Intra has a gain of around 0.9 dB over JPEG2000 for *Crew*. In our case, the gain is around 0.1 dB in favor of AVC HP Intra. For the three sequences mentioned previously, JPEG2000 outperforms AVC MP Intra for the *Harbour* and *Crew* sequences with gains around 0.1∼1.0 and 0.5 dB in PSNR respectively. Further, it performs as well as AVC MP Intra for the *City* sequence. Thus, JPEG2000 globally outperforms AVC MP Intra and is very close to AVC HP Intra in RD performance. The *Soccer* sequence confirms this as the performance of JPEG2000 is very close to AVC HP Intra, around 0.1 dB difference, and outperforms AVC MP Intra with around 0.6 gain in PSNR. This difference in RD performance compared to the same sequences in [59] and the *Harbour* sequence in [57] is due to the fact that [57, 59] use the Visual Frequency Weighting for the JPEG2000 encoder, which results in a drop in PSNR RD performance for JPEG2000. In [58], JPEG2000 is very close to AVC HP Intra for a set of monochromatic still images with a much higher resolution up to 2048x3072. Finally, JPEG XR is inferior to JPEG2000 by around 0.5∼0.8 dB except for *Crew* where it has a similar performance to JPEG2000 and AVC HP Intra. As for 1080p spatial resolution, JPEG is significantly outperformed by the remaining codecs with a maximum gap around 4.0 dB with respect to AVC HP Intra.

352x288 (CIF) spatial resolution

AVC Intra in both profiles outperforms JPEG2000 as illustrated in Figures 3.14 and 3.15. The gain is around 0.5∼1.0 dB for the High Profile and around 0.1∼0.5 for the Main Profile. Nevertheless, the difference between AVC Intra and JPEG2000 is small for sequences with more or less uniform texture such as *Crew*, *Harbour*, *Crew*, *Football* and *Coastguard*. Furthermore, JPEG XR is very competitive with JPEG2000 as they have a similar performance for most sequences except for *Football*, *Hallmonitor* and *Foreman* where it is outperformed by JPEG2000 by around 0.5∼1.0 dB. On the other hand, JPEG XR is better in RD performance than JPEG2000 for *Ice* and *Mobile*.

176x144 (QCIF) spatial resolution

Globally, the Main and High Profiles have a similar RD performance since the High Profile has been introduced to improve the performance for high spatial resolutions. This is observed in Figures 3.16 and 3.17. Furthermore, AVC Intra has a gain of 1.0∼2.0 dB in PSNR over JPEG2000. JPEG XR has a similar performance to AVC HP Intra for *Mobile* otherwise its performance is situated between AVC Intra and JPEG2000. It outperforms JPEG2000 by around 1.0 dB for *Foreman* and *Ice*. For the remaining sequences, JPEG2000 and JPEG XR have a similar performance.
Finally, compared to [57, 58], in addition to the Visual Frequency Weighting mentioned previously, different wavelet decomposition levels are used for the different spatial resolutions. On the other hand, a fixed number of 5 decomposition levels, which is not always optimal, is used in [57–59].



(a) *Akiyo*.

(b) *Bus*.

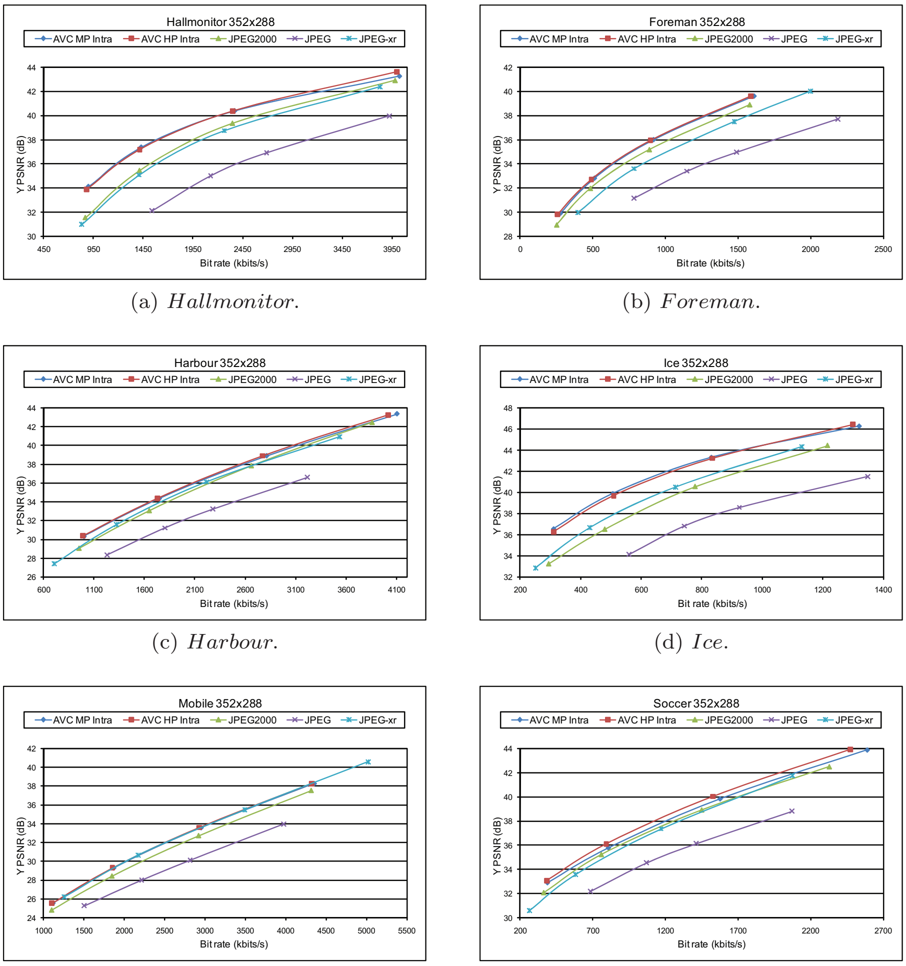
(c) *City*.

(d) *Coastguard*.

(e) *Crew*.

(f) *Football*.

Figure 3.14: RD performance for CIF spatial resolution.



(a) *Hallmonitor*.

(b) *Foreman*.

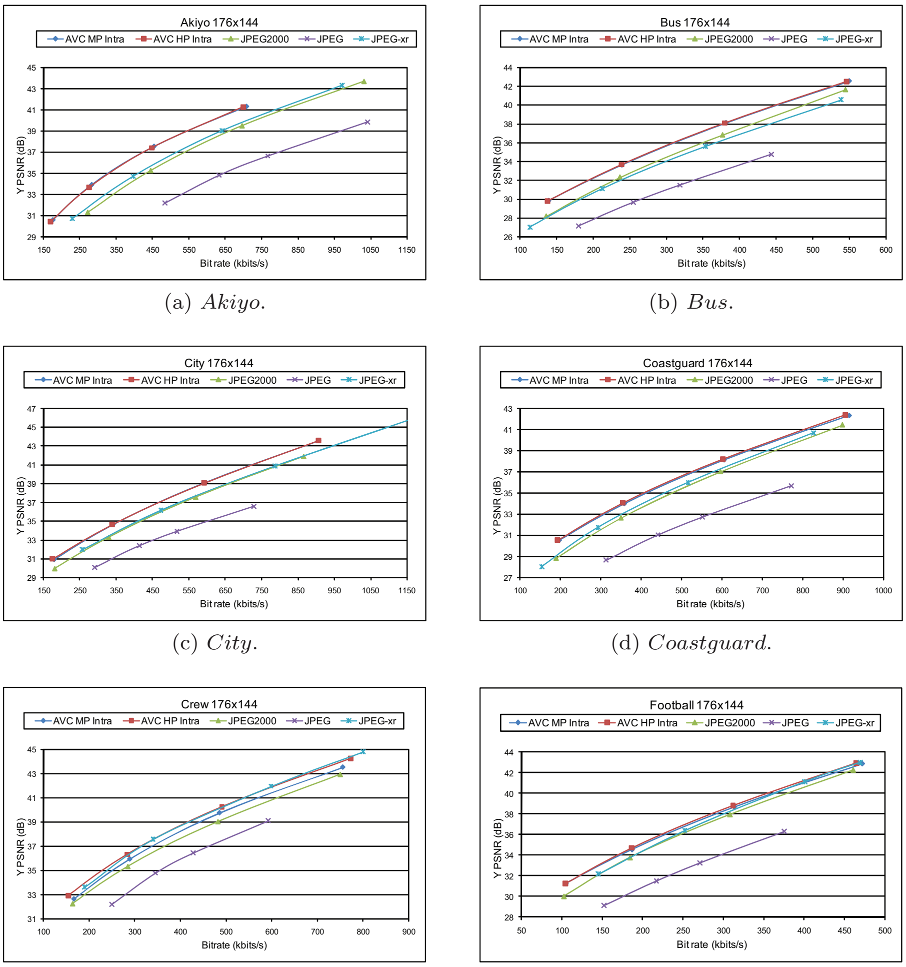
(c) *Harbour*.

(d) *Ice*.

(e) *Mobile*.

(f) *Soccer*.

Figure 3.15: RD performance for CIF spatial resolution.



(a) *Akiyo*.

(b) *Bus*.

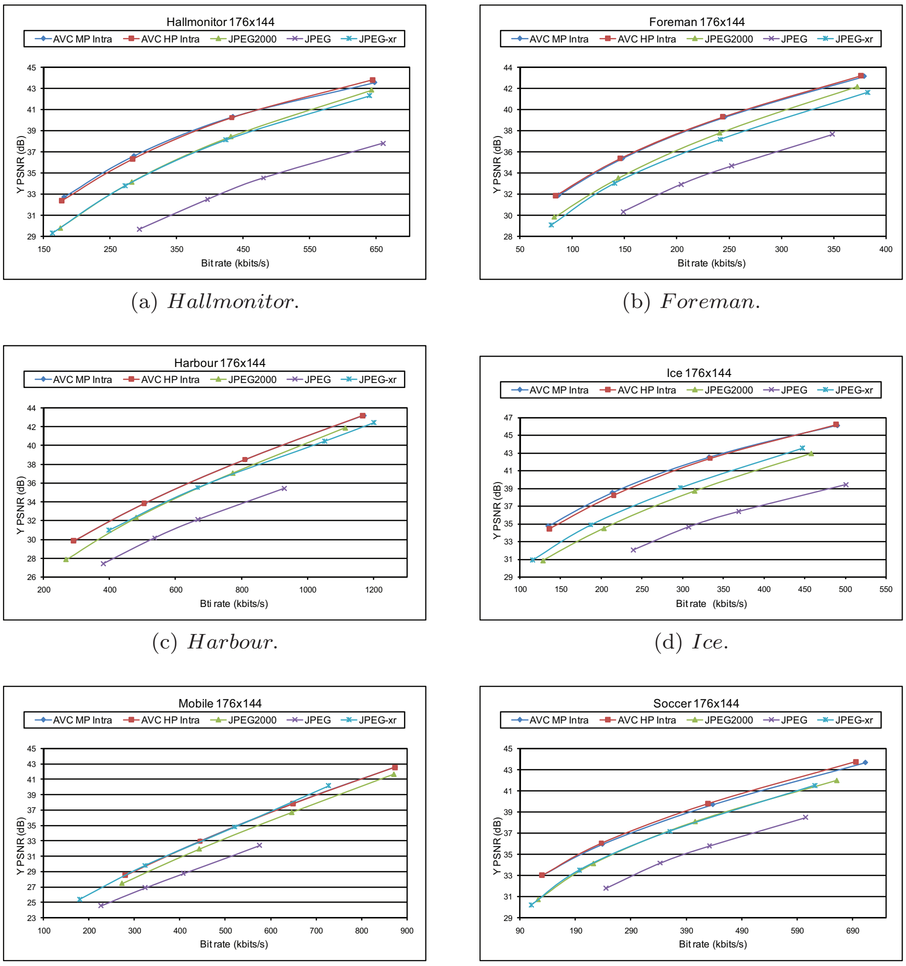
(c) *City*.

(d) *Coastguard*.

(e) *Crew*.

(f) *Football*.

Figure 3.16: RD performance for QCIF spatial resolution.



(a) *Hallmonitor*.

(b) *Foreman*.

(c) *Harbour*.

(d) *Ice*.

(e) *Mobile*.

(f) *Soccer*.

Figure 3.17: RD performance for QCIF spatial resolution.

3.7 Conclusion

In this chapter, an evaluation of several codecs for Intra frame encoding of video is performed using a rich set of video sequences. The widely used PSNR is chosen as the distortion metric. Furthermore, the parameters of each codec are fine-tuned to obtain the best performance out of it, especially for JPEG2000 and AVC Intra.

For high spatial resolution sequences greater than 704x576, the simulation results show that JPEG2000 is very competitive with AVC High Profile Intra and outperforms JPEG XR. For intermediate and low spatial resolution sequences JPEG2000 is outperformed by both profiles of AVC Intra and has a similar performance to JPEG XR. Thus, JPEG2000 is interesting for applications with high resolution video such as satellite and medical imaging and digital cinema. In addition, JPEG2000 provides some additional interesting features such as scalability, Region of Interest definition and rate control that AVC and JPEG XR do not provide. This is in addition to the royalty fee free, license fee free nature of the JPEG2000 standard.

The evaluation methodology used to compare compression schemes is of a primary importance. In this work, a simple evaluation methodology is considered, which consisted in comparing the RD characteristics of each codec under study for a number of video sequences with different resolutions. It would be interesting to consider other metrics than PSNR that take into consideration the Human Visual System. Furthermore, the sequences are simply clustered with respect to the spatial resolution. One could perform a more sophisticated clustering such as the nature of the video content and then perform the evaluation. A more appropriate evaluation methodology would be to compare also the visual quality of the decoded video produced by different codecs. Finally, a comparison of complexity and memory requirements between the codecs under study in this research should be performed in order to produce a better understanding of their relative performance.

Chapter 4

Monoview DVC

4.1 Introduction

The key task of exploiting the source statistics in DVC is carried out in the SI generation process to produce an estimation of the WZ frame at the decoder. SI has a significant influence on the RD performance of DVC. Indeed, more accurate SI implies that fewer bits are requested from the encoder so that the bit rate is reduced for a similar quality. In common DVC codecs, the SI is obtained by MCTI from the previous and the next key frames and uses the Block Matching Algorithm (BMA) for motion estimation. However, motion vectors from BMA are often not faithful to true object motion. Unlike classic video compression, it is more important to find true motion vectors for SI generation in DVC. Therefore, it is important to improve the SI generation process in DVC in order to achieve a better RD performance. In contrast to the motion compensation techniques used in conventional codecs, MCTI has no knowledge about the frame being decoded. MCTI holds as long as the block has constant velocity. However, MCTI fails to generate a good SI estimation when the motion is large or asymmetric.

In this chapter, a new SI generation scheme by exploiting spatio-temporal correlations at the decoder is proposed. It uses the Partially Decoded WZ (PDWZ) frame generated by the WZ decoder to improve the SI generation. In other words, the proposed scheme is not only based on the key frames, but also on the WZ bits already decoded. Furthermore, an enhanced temporal frame interpolation is applied, including refinement and smoothing of the motion vectors, and optimal compensation mode selection. The latter selects the mode to with the minimum matching distortion.

Another appealing property of DVC is its good resilience to transmission errors due to its intrinsic joint source-channel coding framework. A thorough analysis of its performance in the presence of transmission errors has been presented in [19, 20], showing its good error resilience properties. This results from the fact that DVC is based on a statistical framework rather than the closed-loop prediction used in conventional video coding. Recently, the rapid growth of Internet and wireless communications has led to increased interest for robust transmission of compressed video. However, transmission errors may severely impact video quality as compressed data is very sensitive to these errors [78].

Thus, error control techniques are necessary for efficient video transmission over error-prone channels.

Error concealment consists in estimating or interpolating corrupted data at the decoder from the correctly received information. Moreover, It improves the quality of the decoded video corrupted by transmission errors, without any additional payload on the encoder or the channel. EC can be classified into three categories, temporal concealment, spatial concealment, and hybrid spatio-temporal concealment [79].

Further, a new hybrid, spatio-temporal, EC scheme for WZ frames is also proposed in this chapter. It uses results from spatial EC to improve the performance of temporal EC, instead of simply switching between spatial and temporal EC. Spatial EC, based on an edge directed filter [15], is initially applied to the corrupted frame. Then, the spatially concealed frames is used as a PDWZ frame to improve the performance of the temporal EC. In other words, the temporal EC is not only based on the key frames, but also on the WZ bits already decoded.

The remaining of this chapter is organized as follows. Initially, work related to improved SI and EC from literature is summarized in section 4.2. Further, the proposed improved SI technique is introduced in section 4.3 followed by the hybrid EC technique in section 4.4. Then, these techniques are evaluated in section 4.5. Finally, conclusions about the improved SI and EC for DVC are drawn in section 4.6.

4.2 Related Work

4.2.1 Improved Side Information

Spatial motion vector smoothing is proposed to improve the performance of bi-directional MCTI in [24, 36]. It is observed that the motion vectors have sometimes low spatial coherence. Therefore, the spatial smoothing aims at reducing the number of false motion vectors, i.e. incorrect motion vectors when compared to the true motion field. This scheme uses weighted vector median filters, which maintain the motion field's spatial coherence. At each block, the filtering looks for a candidate motion vector at the neighboring blocks that minimizes a weighted difference with respect to its neighbors. This filter is also adjusted by a set of weights controlling the smoothing strength depending on the prediction error of the block for each motion vector candidate. However, spatial motion smoothing is only effective at removing false vectors that occur as isolated impulsive noises.

Sub-pixel interpolation is also proposed to improve motion estimation for SI generation in [80]. The sub-pixel interpolation method of AVC/H.264 is used to generate the pixel value at the sub-pixel precision. At the decoder, the SI is motion compensated and refined according to the selected mode from backward, forward and bi-directional modes. This motion refinement procedure uses sub-pixel interpolation to improve the precision of the search, which is effective at improving the generated SI. On the other hand, it fails in the presence of large or asymmetric motion. Moreover, it significantly increases the complexity of the decoder.

Encoder aided motion estimation to improve SI generation is proposed to conduct more accurate motion estimation at the decoder with the help of auxiliary information sent by the encoder, such as CRC [5] and hash [26] bits.

In [5], CRC bits for every block are calculated at the encoder and transmitted to the decoder to perform motion search by choosing the candidate block that produces the same CRC. The encoder transmits a CRC check of the quantized syndrome bits for each block. Motion estimation is carried out at the decoder by searching over the space of candidate predictors one-by-one to decode a sequence from the set labeled by the syndrome. When the decoded sequence matches the CRC check, decoding is declared to be successful. However, the way to generate and exploit CRC is complicated, and it increases the complexity not only at the decoder side, but also at the encoder side.

In [26], robust hash codewords are sent by the encoder, in addition to the WZ bits, to help the decoder in estimating the motion and generating the SI. These hash bits carry the motion information to the decoder without actually estimating the motion at the encoder. The robust hash code for a block consists of a very coarsely subsampled and quantized version of the block. The decoder performs a motion search based on the hash to generate the best SI block from the previous frame. The hash bits do help in motion estimation, however, they increase the encoder's complexity and transmission payload. In addition, the SI is generated based only on the previous key frame in [26], which is not as good as bi-directional motion estimation.

The WZ frame is split into two subsets at the encoder based on a checkerboard pattern in [81], where each one is encoded separately. At the decoder, the first subset is decoded using the SI obtained by MCTI, thus exploiting only temporal correlation. Then, the second subset is decoded either using MCTI or interpolated data from the first decoded subset. More specifically, MCTI is used when the estimated temporal correlation is high. Otherwise, the spatial SI is used. However, this approach can only achieve a modest improvement.

The idea of iterative decoding and motion estimation is proposed to improve SI, such as motion vector refinement via bitplane refinement [82] and iterative MCTI techniques [83, 84], but with a high cost of several iterations of motion estimation and decoding. In [82], the reconstructed frame and the adjacent key frames are used in order to refine the motion vectors and thus obtain an improved version of the SI and the decoded frames. This includes a matching criteria function to perform motion estimation and three decoding interpolation modes to select the best reference frame. This scheme is based on bitplane refinement for pixel-domain DVC and only minor performance improvements have been achieved. In [83], the first outcome of the DVC decoder is called a partially decoded picture. A second motion compensated interpolation is applied, which uses the partially decoded picture as well as the previous and the next key frames. For each aligned block in the partially decoded picture, the most similar block is searched for in the previous frame, the next key frame, the motion compensated average of both frames and the initial MCTI.

An iterative approach based on multiple SI with motion refinement is also proposed in [84]. Multiple SI streams are used at the decoder, the first SI stream is predicted by motion extrapolating the previous two closest key frames and the second SI stream is predicted using the immediate key frame and the closest WZ frame. Based on the error probability, the turbo decoder decides which SI stream is used for decoding a given block.

4.2.2 Error Concealment (EC)

Spatial EC consists in interpolating the lost block from its spatially neighboring available blocks or coefficients within the same frame. It relies on the inherent spatial smoothness of the data.

For example, the technique proposed in [85] exploits the smoothness property of image signals and recovers the damaged blocks by a smoothness measure based on second order derivatives. More precisely, the lost block is estimated such that the reconstructed frame minimizes the smoothness measure between the estimated and the boundary pixels. However, this smoothness measure still leads to blurred edges in the recovered frame due to the simple second order derivative based measure to represent the edges. Some approaches are proposed to minimize variations along edge directions or local geometric structures [86, 87]. However, these schemes require accurate detection of spatial features and mistakes can yield annoying artifacts.

In [88], through benchmarking results on the existing EC approaches, it is observed that none of the existing approaches is an all time champion. Therefore, a classification based concealment approach is proposed which can combine different spatial approaches. Block classification is also proposed to take advantage of various concealment algorithms by adaptively selecting the suitable algorithm for each block [89].

Temporal EC techniques use the temporally neighboring frames to estimate the lost blocks in the current frame based on the assumption that the video content is smooth and continuous in the temporal domain. A very simple scheme for temporal EC is to just copy the block at the same spatial location in the previous frame to conceal the lost block. If the motion vector is not damaged or lost, the motion compensated prediction from the reference frame can also be used to conceal the damaged block.

A temporal EC algorithm is proposed in [90] which selects the motion vector with the minimum side match distortion among multiple reference frames. EC with multiple reference frames is used in [90] to improve the performance of the frame reconstruction. This scheme can often produce good results, especially when there is little irregular motion in the scene. The problem, however, is that the block and the coding mode (Intra or Inter) may not be available when a packet is lost.

Other approaches focus on the recovery of the lost motion vectors for temporal concealment [91, 92]. For example, a bi-directional temporal EC algorithm that can recover from the loss of a whole frame is proposed in [91]. However, the accuracy of the motion estimation may significantly affect the results.

Temporal EC usually leads to better results when compared to spatial concealment given the typically high temporal correlation in video. However, for video with scene changes or with very large or irregular motion, spatial EC is preferred.

Some attempts have been made to combine both spatial and temporal EC in [93, 94]. These schemes use some mode selection methods to decide whether to use spatial or temporal EC. For example, temporal activity (measured as the prediction error in the surrounding blocks) and spatial activity (measured as the variance of the same surrounding blocks) are used to decide which concealment mode to use [93]. In general, these methods have achieved very limited success, mostly due to the simple mode selection mechanisms at the decoder to merge

the results from both concealment.
In [95], a forward error correcting coding scheme is proposed for conventional video coding, where an auxiliary redundant bitstream generated at the encoder using WZ coding is sent to the decoder for EC.
However, in the literature, there are no EC schemes for DVC. Most probably, this is the first attempt to propose concealment techniques for DVC.

4.3 Proposed Improved Side Information

The DVC decoder's architecture including the proposed SI generation scheme is illustrated in Figure 4.1. First, the MCTI with spatial motion smoothing from [36] is used to compute the Initial SI (ISI) for the frame being decoded. Based on the ISI, the WZ decoder is first applied to generate the PDWZ, which is then exploited to generate an improved SI as detailed in Figure 4.2. More specifically, the SI refinement procedure first detects suspicious motion vectors based on the matching errors between the PDWZ frame and the reference key frames. These motion vectors are then refined using a new matching criterion and a spatial smoothing filter. Furthermore, optimal motion compensation mode selection is conducted. Namely, the interpolated block can be selected from a number of sources: the previous frame, the next frame, and the bi-directional motion-compensated average of both frames. The Final SI (FSI) is constructed using motion compensation based on the refined motion vectors and the optimal compensation mode. Finally, based on the FSI, the reconstruction step is performed again to get the final decoded WZ frame.

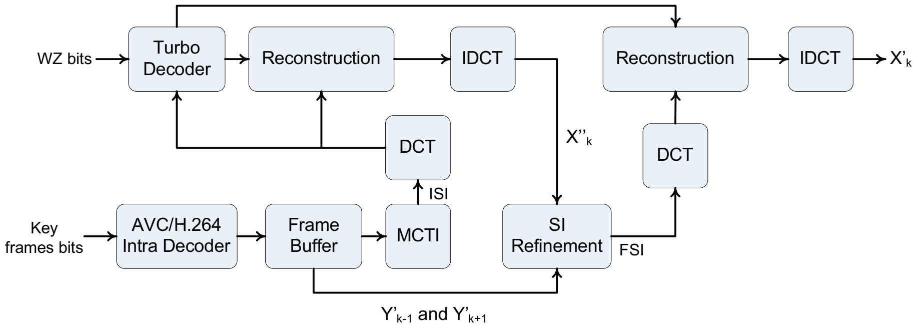


Figure 4.1: DVC decoder architecture with proposed SI generation.

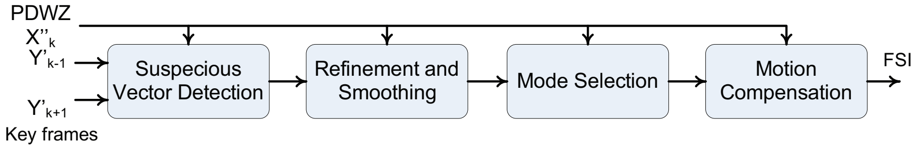


Figure 4.2: Proposed SI refinement procedure.

This is expected to be efficient in situations where motion is significant as the difference in prediction quality between the ISI and FSI is more significant. The reason for this is that the final SI is highly correlated with the WZ frame in the case of high activity video content. Therefore, most of the SI values map into the decoded bin in the reconstruction process (i.e. the SI value is taken as the reconstructed value). This would produce a better reconstruction with lower distortion as less SI values are truncated into the quantization interval. On the contrary, more SI values are truncated in the initial reconstruction phase, as the initial SI is less correlated with the WZ frame, producing a more distorted reconstruction.

On the other hand, the improvement for low motion video is less significant as both side informations, initial and final, are close in terms of prediction quality. Common MCTI techniques use only the previous and next key frames to generate the SI. In comparison, the proposed SI generation scheme appears to perform much better than common MCTI, since it has additional information (from WZ bits) about the frame it is trying to estimate. Moreover, the spatio-temporal correlations are exploited based on the PDWZ frame using the SI refinement procedure. The decoded frame obtained here could then be used again as PDWZ frame for a subsequent iteration. However, our experiments show that extra iterations do not provide a significant performance improvement. In other words, the additional information carried by parity bits is fully exploited in a single run of our SI generation scheme. Therefore, only one iteration is used in the proposed scheme, avoiding additional complexity at the decoder.

4.3.1 Matching Criterion

To exploit the spatio-temporal correlations between the PDWZ frame and reference key frames, a new matching criterion is used to evaluate the error in motion estimation. Generally, the goal of motion estimation is to minimize a cost function that measures the prediction error for a given block such as the popular Mean Absolute Difference (MAD) defined as

$$MAD(P_0, F_1, F_2, mv) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |F_1(i+x_0+x, j+y_0+y) - F_2(i+x_0, j+y_0)|, \tag{4.1}$$

where (x_0,y_0) is the coordinate of the top left point P_0 of the original block in the current frame F_1, F_2 the reference frame, (x,y) the estimated motion vector mv, and (M,N) the dimensions of the block. However, when there are changes in pixel intensity and noise, minimizing MAD often leads to false motion vectors. On the other hand, Boundary Absolute Difference (BAD) is proposed in the error concealment literature [92, 96]. BAD measures the accuracy of motion compensation to enforce the spatial smoothness property by minimizing the side matching distortion between the internal and external border of the recovered block. It is defined as

$$
\begin{aligned}
&BAD(P_0, F_1, F_2, mv) = \\
&\frac{1}{M}\sum_{i=0}^{M-1}(|F_1(i+x_0, y_0) - F_2(i+x_0+x, y_0+y-1)|) + \\
&\frac{1}{M}\sum_{i=0}^{M-1}(|F_1(i+x_0, y_0+N-1) - F_2(i+x_0+x, y_0+y+N)|) + \\
&\frac{1}{N}\sum_{J=0}^{N-1}(|F_1(x_0, j+y_0) - F_2(x_0+x-1, j+y_0+y)| + \\
&\frac{1}{N}\sum_{J=0}^{N-1}(|F_1(x_0+M-1, j+y_0) - F_2(x_0+x+M, j+y_0+y)|). \qquad (4.2)
\end{aligned}
$$

Unfortunately, BAD is not efficient at picking out bad motion vectors when local variation is large [96].
In this research, a new matching criterion based on MAD and BAD is proposed. The matching distortion (DST) between two frames is defined as

$$DST(P_0, F_1, F_2, mv) = \alpha BAD(P0, F_1, F_2, mv) + (1-\alpha)MAD(P_0, F_1, F_2, mv), \qquad (4.3)$$

where α is a weighting factor. MAD is utilized to measure how well the candidate mv can keep temporal continuity. The smaller MAD is, the better the candidate mv keeps temporal continuity. On the other hand, BAD is used to measure how well the candidate mv can keep spatial continuity. The smaller BAD is, the better the candidate mv keeps spatial continuity.
This new matching criterion is used in suspicious vector detection, motion vector refinement and smoothing, and optimal motion compensation mode selection in the proposed improved SI generation pipeline as detailed further.

4.3.2 Suspicious Vector Detection

Generally, for most sequences with low and smooth motion, the majority of motion vectors estimated by MCTI are close to the true motion. However, erroneous vectors may result in serious block artifacts if they are directly used in frame interpolation. Therefore, a threshold T is established to define the candidate blocks for further refinement based on the matching criterion DST. If an estimated mv satisfies the criteria defined in Equation 4.4, it is considered to be a good estimation. Otherwise, it is identified as a suspicious vector and will be further processed.

$$DST(P_0, Y'_{k-1}, X''_k, mv) + DST(P_0, Y'_{k+1}, X''_k, mv) < T, \qquad (4.4)$$

where Y'_{k-1} and Y'_{k+1} are the previous and next decoded key frame, respectively, and X''_k is the PDWZ frame.

4.3.3 Motion Vector Refinement and Smoothing

The spatio-temporal correlations between the PDWZ frame and the reference key frames are exploited to refine and smooth the estimated motion vectors.

More specifically, a spatial motion smoothing filter is therefore used, similar to [24], but with the matching criterion defined in Equation 4.3 and the PDWZ frame. More precisely, a weighted median vector filter is used to maintain the motion field spatial coherence. This filter is adjusted by a set of weights controlling the smoothing strength. The weighted median vector filter is defined as

$$mv_f = argmin_{mv_i} \sum_{j=1}^{Num} w_j ||mv_i - mv_j||_L, i \in [1, Num], \tag{4.5}$$

where mv_1, ..., mv_{Num} are the motion vectors of the corresponding nearest neighboring blocks. mv_f is the motion vector output of the weighted vector median filter, which is chosen in order to minimize the sum of distances (L^2-norm) to the other neighboring motion vectors. 8-neighborhood is used in this research (Num=8), as shown in Figure 4.3. The weights w_1, ..., w_{Num} are calculated based on the new matching criterion and the PDWZ frame such as

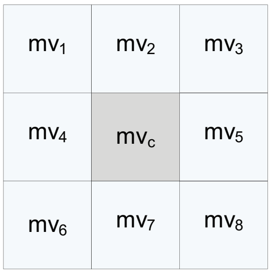


Figure 4.3: Neighboring motion vectors for weighted median vector filter.

$$w_j = \frac{D_{ST}(P_0, Y'_{k-1}, X''_k, mv_c) + D_{ST}(P_0, Y'_{k+1}, X''_k, mv_c)}{D_{ST}(P_0, Y'_{k-1}, X''_k, mv_j) + D_{ST}(P_0, Y'_{k+1}, X''_k, mv_j)}, \tag{4.6}$$

where mv_c is the candidate motion vector for the block to be interpolated. The weight is small if there is a high prediction error using this vector, i.e., the median filter is to substitute the previously estimated motion vector with a neighboring vector which has the smallest prediction error.

4.3.4 Optimal Motion Compensation Mode Selection

The goal of this step is to generate an optimal motion compensated estimate. In most DVC schemes, while bi-directional prediction is shown to be effective, it is limited to motion-compensated average of the previous and the next key frames.

Based on the PDWZ frame, the most similar block to the current block can be selected from three sources: the previous frame, the next frame or the motion-compensated average of both frames. More specifically, the block is estimated by selecting the mode with minimum matching error from the following three modes of motion compensation:

- **Backward mode** - The block in the SI is interpolated using only one block from the previous key frame.

- **Forward mode** - The block in the SI is interpolated using only one block from the next key frame.
- **Bi-directional mode** - The block in the SI is interpolated using the average of one block from the next key frame and another block from the previous key frame.

Among these modes, the decision is performed according to the matching criterion defined in Equation 4.3 as the one with the minimum matching error is retained.
Based on the refined motion vectors and the selected interpolation mode, motion compensation is applied to generate the final SI.

4.4 Hybrid Error Concealment (EC)

The proposed technique used to improve SI generation for DVC not can only improve the performance of DVC, but is also useful to improve the error resilience of DVC when applied to a hybrid error concealment scheme. A hybrid error concealment scheme is proposed based on the improved SI generation technique as illustrated in Figure 4.4.

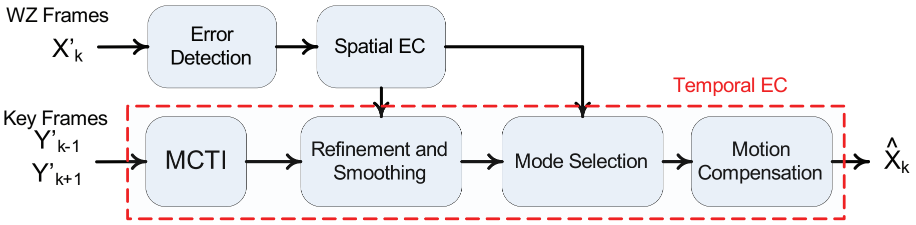
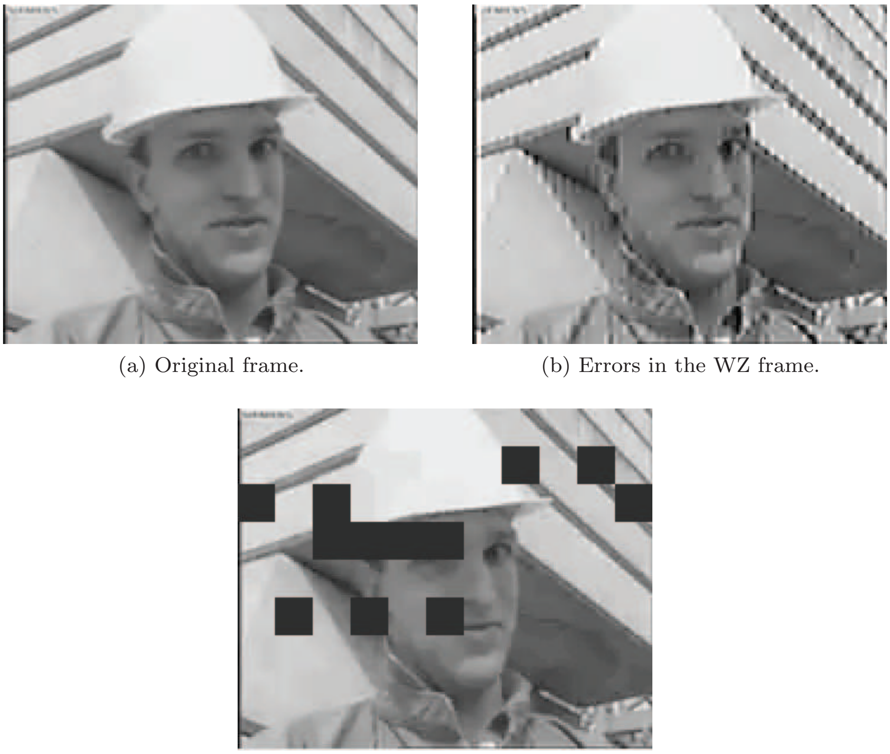


Figure 4.4: Proposed spatio-temporal EC.

The error location is firstly detected. In this research, the error locations are assumed to be known at the decoder, as often presumed in error concealment literature, which can be done at the transport level or based on syntax and watermarking [78]. Spatial EC is then applied to obtain a partially error-concealed frame, which is much closer to the error-free frame than the corrupted one. The partially error-concealed frame is used for motion vector refinement and smoothing, and optimal compensation mode selection to obtain an estimate of the motion vector for the corrupted block. Motion compensation is finally used to obtain the concealed block.

4.4.1 Spatial Error Concealment Based on an Edge Directed Filter

In DVC, the decoded WZ frames are based on the SI generated by MCTI of the key frames. The SI is then used by the decoder, along with the WZ bits, to obtain the decoded WZ frame. The transmission errors in WZ bits tend to cause noises around edges in the corrupted WZ frames. For example, when there are errors in WZ bits, the error pattern of the damaged WZ frames, as shown in



(a) Original frame.

(b) Errors in the WZ frame.

(c) Errors in AVC/H.264 Intra frame.

Figure 4.5: The visual effect of packet losses on WZ and conventionally coded frames.

Figure 4.5(b), is different from those of traditional video coding schemes (Figure 4.5(c)). Therefore, the error concealment schemes proposed for traditional video coding schemes cannot be directly applied to conceal errors in WZ frames. Therefore, based on the error pattern of the WZ frames, an edge directed filter is constructed to remove the noises around edges caused by errors.
Anisotropic diffusion techniques are widely used in image processing for their efficiency at smoothing noisy images while preserving sharp edges. The anisotropic diffusion is adopted as a direction diffusion operation and use the diffusion function for spatial EC as in [15]. The EC method in [15] is designed for wavelet-based images and contains wavelet domain constraints and rectifications. In our case, the edged directed filter is used without any constraint or rectification to remove the noises around edges. The diffusion function is defined as follows

$$f(\nabla I) = \frac{exp(\frac{-|\nabla I|}{M})}{max(exp(\Delta I), 1 + |\nabla I|)}, \tag{4.7}$$

$$M = max_{P \in \Gamma}(|\nabla I_P|), \tag{4.8}$$

where Γ is the corrupted 16x16 pixels block, ∇ is the gradient operator, $|\nabla I|$ is the magnitude of ∇I and ΔI is the Laplacian of the frame I, a second order derivative of I.

The edge directed filter is applied iteratively such that

$$I^{n+1} = I^n + \frac{\Delta t}{N} \sum_{i=1}^{N} f(\nabla I_i^n) \nabla I_i^n, \tag{4.9}$$

where I_{n+1} is the recovered frame after $n+1$ iterations, and I_0 is the corrupted frame. For each pixel (I_i), the filtering is carried out on the neighboring N (16x16) pixels.

4.4.2 Enhanced Temporal Error Concealment

The performance of temporal EC is improved by using the spatially concealed frame. This process is similar to the proposed improved SI described in section 4.3. Therefore, the partially concealed frame, resulting from the spatial EC, is used to exploit the spatio-temporal correlations between the WZ frame and the reference key frames in a better way. Hence, the performance of the temporal EC is improved.

Frist, the matching criterion in Equation 4.3 is used to evaluate the error in motion estimation based on the partially concealed frame and the reference key frames. Then, the resulting motion vectors are refined and smoothed in the same way as described in section 4.3.3. Further, the concealed block is selected from a number of sources: the previous frame, the next frame, the motion-compensated average of both frames as presented in section 4.3.4. Finally, motion compensation is performed to generate the concealed blocks as a result of the temporal concealment.

4.5 Simulation Results

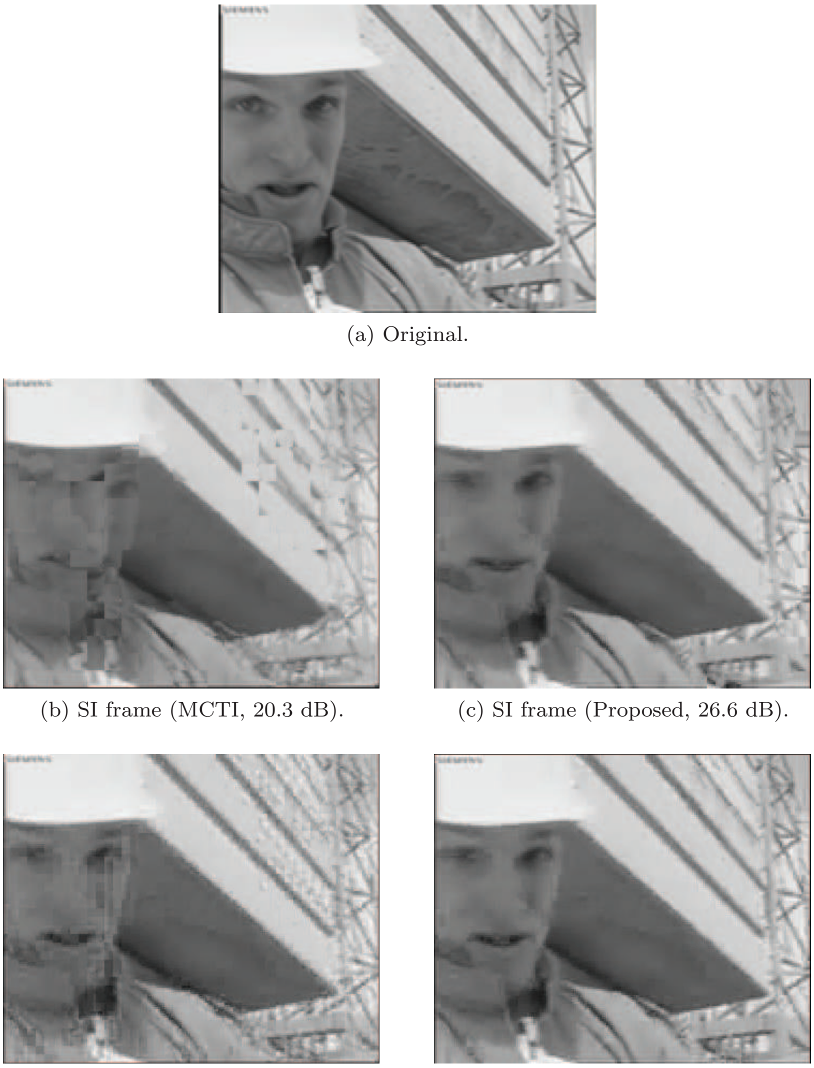
The Transform Domain WZ (TDWZ) DVC codec proposed in [36] is used in our experiments and only luminance data is encoded. The video sequences *Foreman*, *Soccer*, *Coastguard* and *Hallmonitor* are used in the QCIF/CIF format at 15/30 fps.

4.5.1 Improved SI Performance

Figure 4.6 shows a sample frame from the SI and decoded WZ frames for *Foreman.* The face and the building in the SI generated by the TDWZ contains block artifacts (4.6(b)). On the contrary, the SI generated by the proposed method (Figure 4.6(c)) is much better. The improvement in the SI also results in a better reconstruction for the WZ frame. There are much fewer block artifacts on the face and the building in the decoding frame (Figure 4.6(e)) for the proposed method when compared to the TDWZ (Figure 4.6(d)).

Figure 4.7 shows the SI quality for *Foreman.* The proposed algorithm achieves up to 6.7 dB and an average of 2.4 dB improvement when compared to the SI in the TDWZ. The PSNR values of the decoded WZ frames are shown in Figure 4.8. Compared to the TDWZ, the proposed method achieves up to 2.2 dB and an average of 0.6 dB improvement.

The RD performance for all the sequences is shown in Figures 4.9, 4.10, 4.11 and 4.12. A RD performance improvements over the TDWZ are observed for



(a) Original.

(b) SI frame (MCTI, 20.3 dB).

(c) SI frame (Proposed, 26.6 dB).

(d) Decoded frame (MCTI, 27.5 dB).

(e) Decoded frame (Proposed, 29.8 dB).

Figure 4.6: Visual perception of the different SI and decoded frames for *Foreman*.

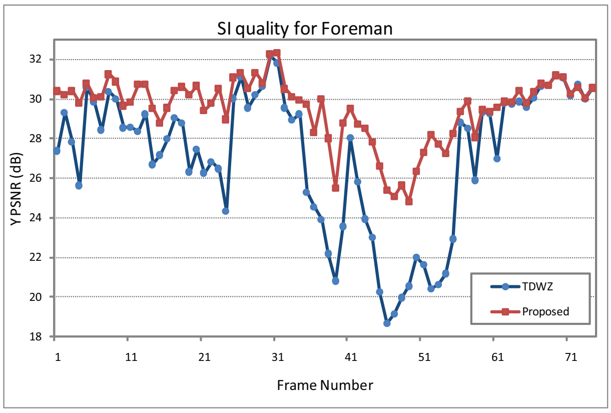


Figure 4.7: PSNR of SI for *Foreman* frames comparing the proposed Improved SI with TDWZ.

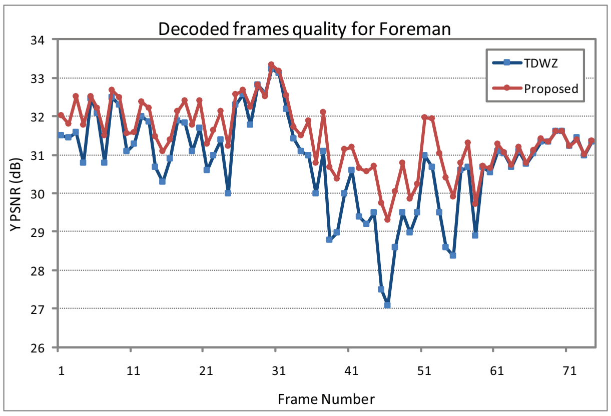


Figure 4.8: PSNR of decoded *Foreman* frames comparing the proposed Improved SI with TDWZ.

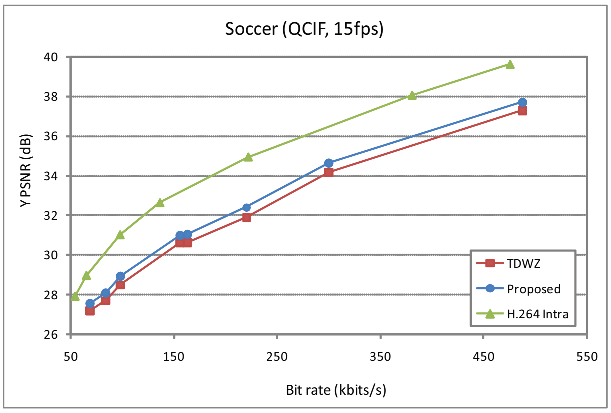


Figure 4.9: RD performance for sequence *Soccer* comparing the proposed Improved SI with TDWZ.

all sequences.

For *Soccer* (Figure 4.9), the proposed method significantly improves the objective quality up to 1.0 dB with respect to TDWZ. When compared to AVC/H.264 Intra, the performance is still inferior but the gap is brought down to around 2.0 dB.

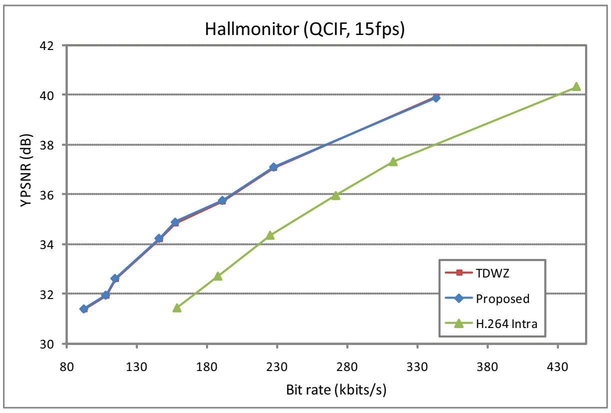


Figure 4.10: RD performance for sequence *Hallmonitor* comparing the proposed Improved SI with TDWZ.

For video with easy motion such as *Hallmonitor* (Figure 4.10), the improvement of the proposed method is negligible when compared to TDWZ, which is already around 3.0 dB superior to AVC/H.264 Intra.

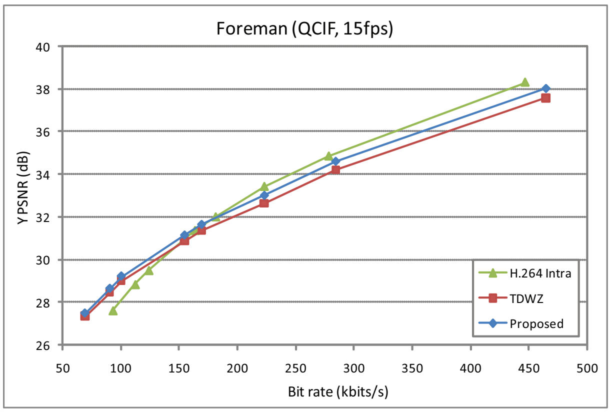


Figure 4.11: RD performance for sequence *Foreman* comparing the proposed Improved SI with TDWZ.

For *Foreman*, the introduced SI improves the performance over AVC/H.264 Intra for low bit rates (Figure 4.11). For *Coastguard*, the introduced SI outperforms AVC/H.264 Intra for all different bit rates (Figure 4.12).

The proposed SI generation is evaluated for *Foreman* and *Soccer* in the CIF format at 30 fps. The corresponding RD performance is depicted in Figure 4.13 for *Foreman* and Figure 4.14 for *Soccer*.

When compared to the QCIF results (Figure 4.9 and 4.11), the improvement

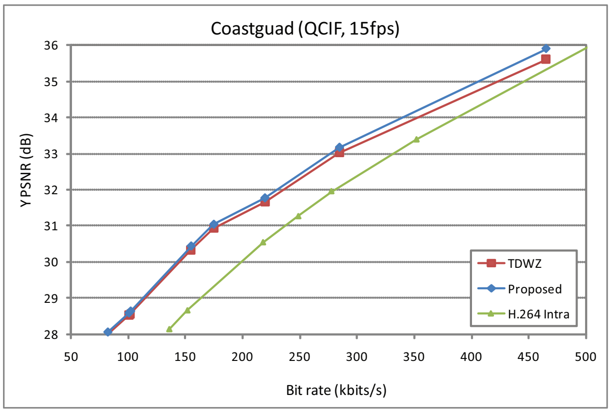


Figure 4.12: RD performance for sequence *Coastgurad* comparing the proposed Improved SI with TDWZ.

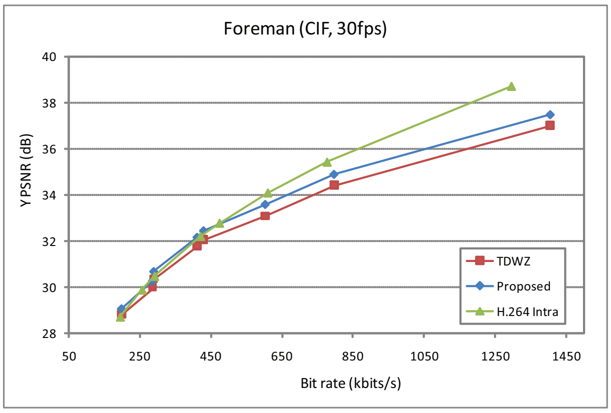


Figure 4.13: RD performance for sequence *Foreman* comparing the proposed Improved SI with TDWZ.

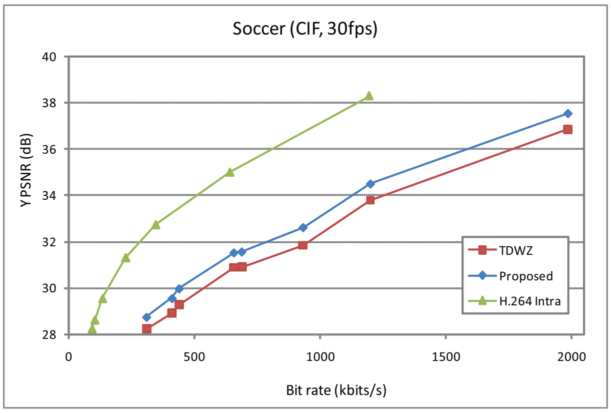


Figure 4.14: RD performance for sequence *Soccer* comparing the proposed Improved SI with TDWZ.

of the proposed method over TDWZ for the CIF format is less significant than that for the QCIF format. The reason is that a larger frame rate is used for the CIF format, which means that the motion between two subsequent frames is smaller. This improves the quality of the SI generated by MCTI in the TDWZ, which makes the gain obtained by the proposed method less significant. All the results above use value 0.3 for the weight α and a threshold T value of 10.

Table 4.1: Effect of the weight α (*Foreman*, QCIF, 15fps) on the decoded video.

Bit rate	Decoded frame PSNR (α=0.1)	Decoded frame PSNR (α=0.3)	Decoded frame PSNR (α=0.5)
69.37 kbps	27.54 dB	27.56 dB	27.51 dB
100.61 kbps	29.11 dB	29.14 dB	28.99 dB
169.47 kbps	31.40 dB	31.45 dB	31.23 dB
284.41 kbps	34.46 dB	34.49 dB	34.16 dB
Average	30.63 dB	30.66 dB	30.47 dB

Table 4.2: Effect of the threshold T (*Foreman*, QCIF, 15fps) on the decoded video. PPB stands for the percentage of processed blocks.

Bit rate (Kbps)	PSNR (dB) T=10	PPB (%) T=10	PSNR (dB) T=40	PPB (%) T=40	PSNR (dB) T=70	PPB (%) T=70
69.37	27.56	85.50%	27.55	24.20%	27.51	12.20%
100.61	29.14	90.80%	29.13	29.00%	29.11	18.40%
169.47	31.45	90.50%	31.44	33.80%	31.42	23.80%
284.41	34.49	94.60%	34.47	33.90%	34.43	25.90%
Average	30.66	90.4%	30.65	30.23%	30.62	20.07%

The effect of these parameters on the finally decoded frames are also studied for *Foreman* and *Soccer*. The average PSNR values of the decoded frames at different bit rates with different α values are shown in Tables 4.1 and 4.3. It is observed that the α value of 0.3 always generates the best quality in PSNR. The average PSNR values and the corresponding percentage of detected suspicious motion vectors with different T are shown in Tables 4.2 and 4.4. The threshold T adjusts the added complexity, where it can be significantly reduced by a large T (e.g. 40) and the same time maintain a similar quality for the decoded frame.

4.5.2 EC Performance

Hybrid EC vs Spatial EC and Temporal EC

In the EC simulations for DVC, a communication channel, characterized by the error pattern files provided in [46] with different packet loss ratios (PLR), is used. The test sequences are corrupted with packet loss rates of 3%, 5%, 10% and 20%. Results are obtained by averaging over ten runs using different error patterns. Furthermore, the encoder is supposed to perform ideal rate control.

Table 4.3: Effect of the weight α (*Soccer*, QCIF, 15fps) on the decoded video.

Bit rate	Decoded frame PSNR (α=0.1)	Decoded frame PSNR (α=0.3)	Decoded frame PSNR (α=0.5)
61.45 kbps	27.11 dB	27.14 dB	27.09 dB
91.19 kbps	28.56 dB	28.57 dB	28.52 dB
163.19 kbps	31.13 dB	31.16 dB	31.12 dB
290.90 kbps	34.20 dB	34.21 dB	34.15 dB
Average	30.25 dB	30.27 dB	30.22 dB

Table 4.4: Effect of the threshold T (*Soccer*, QCIF, 15fps) on the decoded video. PPB stands for the percentage of processed blocks.

Bit rate (Kbps)	PSNR (dB) T=10	PPB (%) T=10	PSNR(dB) T=40	PPB (%) T=40	PSNR (dB) T=70	PPB (%) T=70
61.45	27.14	91.22%	27.13	28.79%	27.10	14.20%
91.19	28.57	92.80%	28.57	32.00%	28.54	19.40%
163.19	31.16	92.50%	31.15	34.60%	31.11	22.80%
290.90	34.21	94.60%	34.18	34.80%	34.13	27.90%
Average	30.27	92.78%	30.26	32.55%	30.22	21.08%

In other words, the number of requested bits for each bitplane for the error free case is determined a priori and used for decoding the corresponding corrupted bitstream. Furthermore, if the bit error probability of the decoded bitplane is higher than 10^{-3}, the decoder uses the corresponding bitplane from the SI. The header of the WZ bitstream, which contains critical information such as frame size, quantization parameters, and GOP size, is assumed as correctly received.

To evaluate the performance of the proposed EC method, first only errors in WZ frames are simulated. Figure 4.15 shows the average PSNR values of the error-concealed frames at several quantization indexes for *Foreman*. The results of the proposed EC are compared to those of spatial concealment (SC, based on edge directed filter [15]), and temporal concealment (TC, based on MCTI [36]). As shown in Figure 4.15, the proposed EC method achieves better objective qualities than both SC and TC, for all packet loss rates and quantization indexes. It is also observed that, although TC generally performs better than SC, the performance of SC can be very close to TC for high video quality.

Figures 4.16 and 4.17 show the results for *Soccer* and *Hallmonitor*, respectively. For sequences with large and complex motion such as *Soccer*, the temporal EC performance can perform worse than that of spatial EC. For sequences with simple motion such as *Hallmonitor*, the performance of the proposed EC is similar to TC.

To evaluate the performance of the proposed method in a more realistic scenario, errors in both key and WZ frames are also simulated. The spatial EC defined in AVC/H.264 JM 11.0 [76] is used to conceal errors in the key frames. Figure 4.18 shows the PSNR values for the EC results at several quantization indexes for *Foreman*. As shown in Figure 4.18, errors in key frames further decrease the performance for all EC techniques, but the proposed EC method leads to even

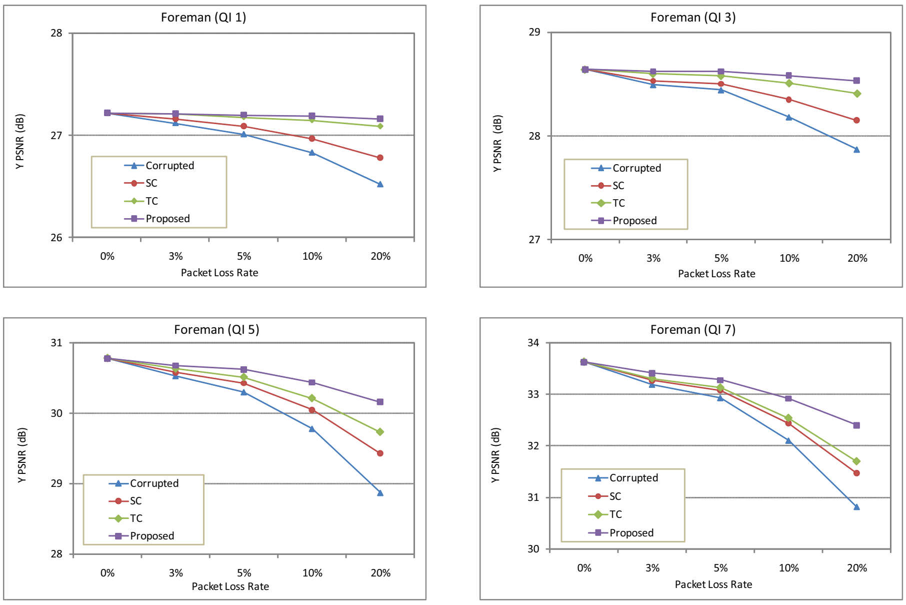


Figure 4.15: PSNR performance for *Foreman* (only WZ frames are corrupted) comparing the different DVC EC methods.

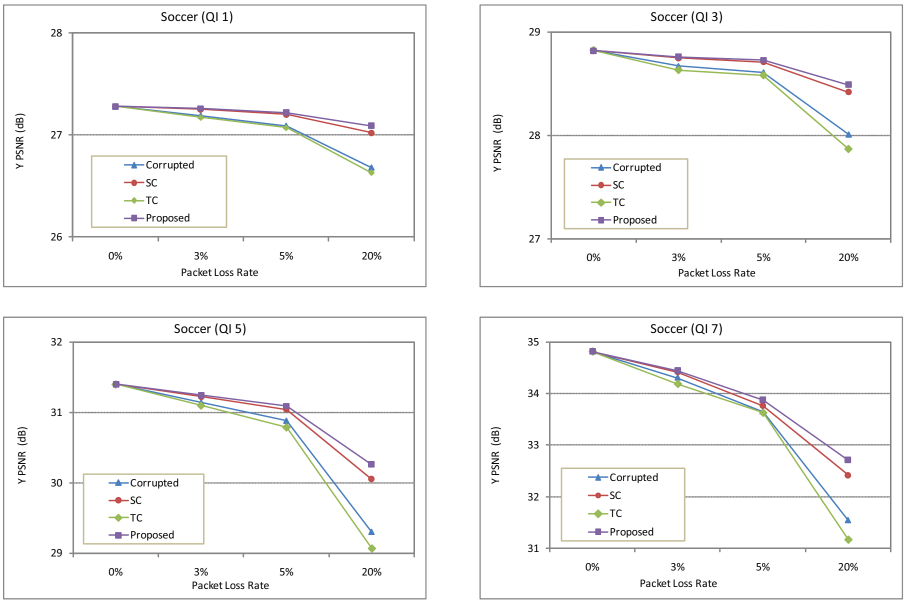


Figure 4.16: PSNR performance for *Soccer* (only WZ frames are corrupted) comparing the different DVC EC methods.

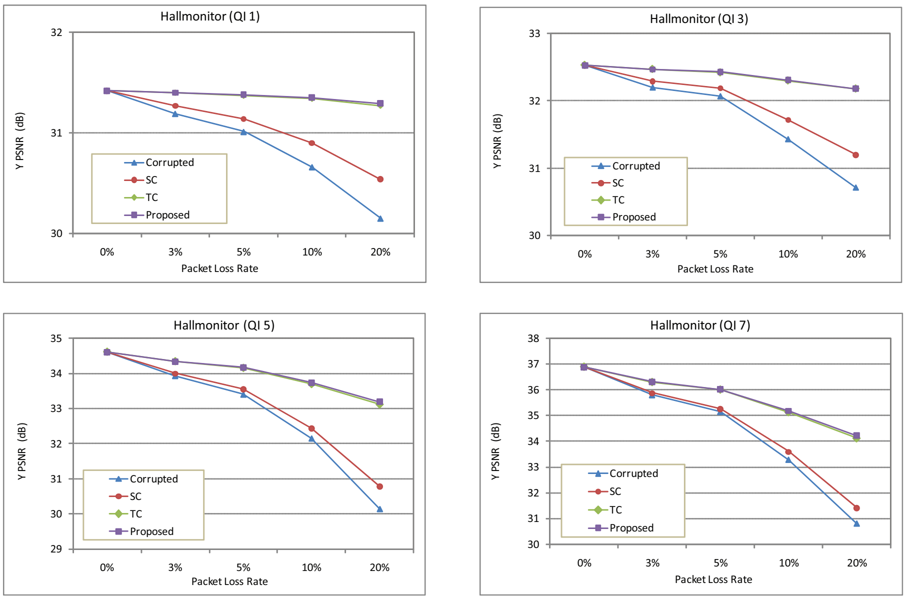


Figure 4.17: PSNR performance for *Hallmonitor* (only WZ frames are corrupted) comparing the different DVC EC methods.

larger improvements. The reason may be that the errors in key frames cause distortions in the WZ frames and these distortions are better concealed by the proposed hybrid EC.
Figure 4.19 shows the PSNR values of all the frames for *Foreman* (QI7, PLR 20%). The improvement by the proposed method is quite important, with gains up to 8.0 dB. The average improvement for the whole sequence is 2.0 dB, significantly outperforming SC or TC.

Figure 4.20 shows the error-concealed frame number 56 in *Foreman* for SC,TC and the proposed EC (QI7, PLR 20%). Distortions are still visible near the edges in the frames concealed by SC (Figure 4.20(b)) and TC (Figure 4.20(c)). The visual quality obtained with the proposed method (Figure 4.20(d)) is better and contains less distortions.

DVC Hybrid EC vs AVC/H.264 EC

Hereafter, The performance of the proposed Hybrid concealment is compared with respect to AVC/H.264 concealed Intra, Inter without motion vectors and Inter with motion vectors [97].
The error concealment algorithm implemented in the AVC/H.264 JM 11.0 [76] reference software is used to conceal the corrupted key frame in DVC. For Intra coding, the AVC/H.264 software Intra error concealment is adopted using spatial interpolation based on weighted average of boundary pixels of the missing block. On the other hand, the error concealment for AVC/H.264 Inter is based on a frame copy method for Inter coding.
Figure 4.21 shows the RD performance for *Foreman* at PLR of 5%, 10%, and 20%. It is noticed that when EC is enabled for DVC and AVC/H.264, at a

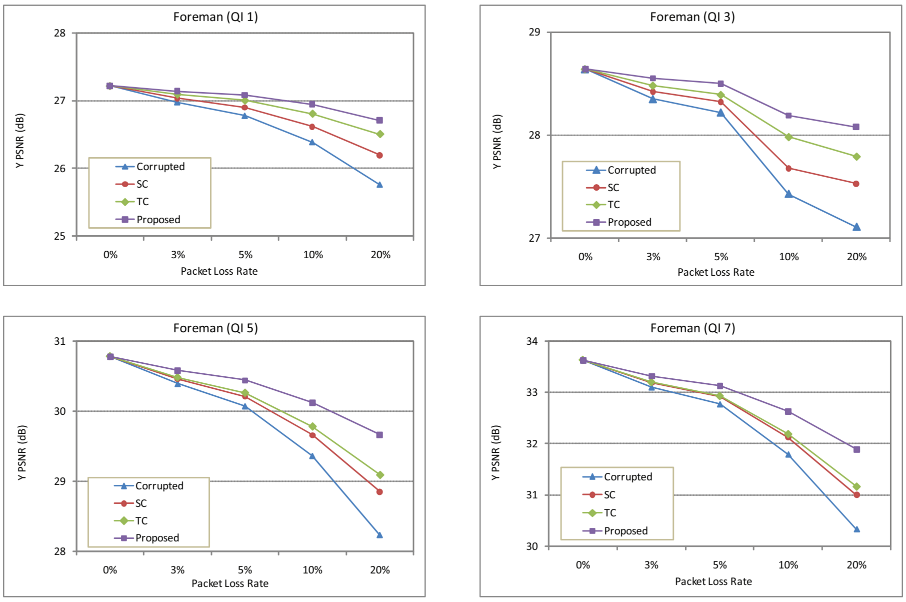


Figure 4.18: PSNR performance for *Foreman* (both key and WZ frames corrupted) comparing the different DVC EC methods.

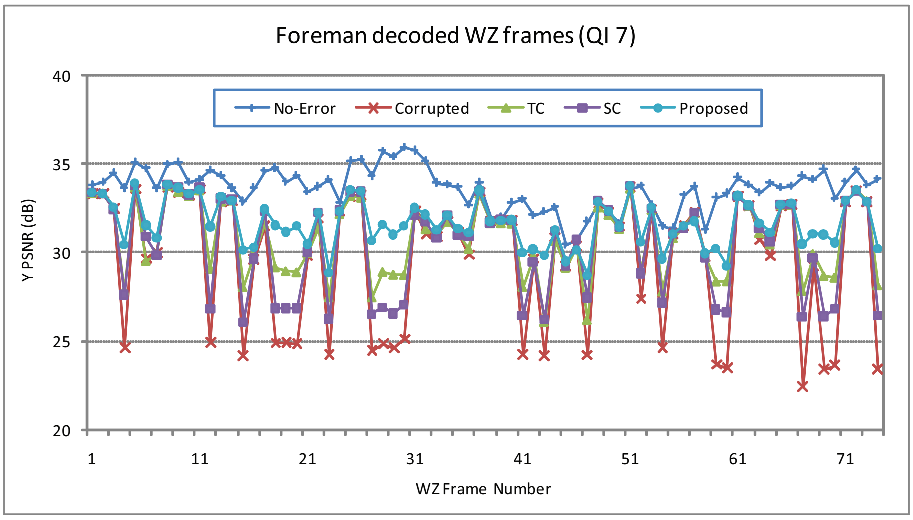
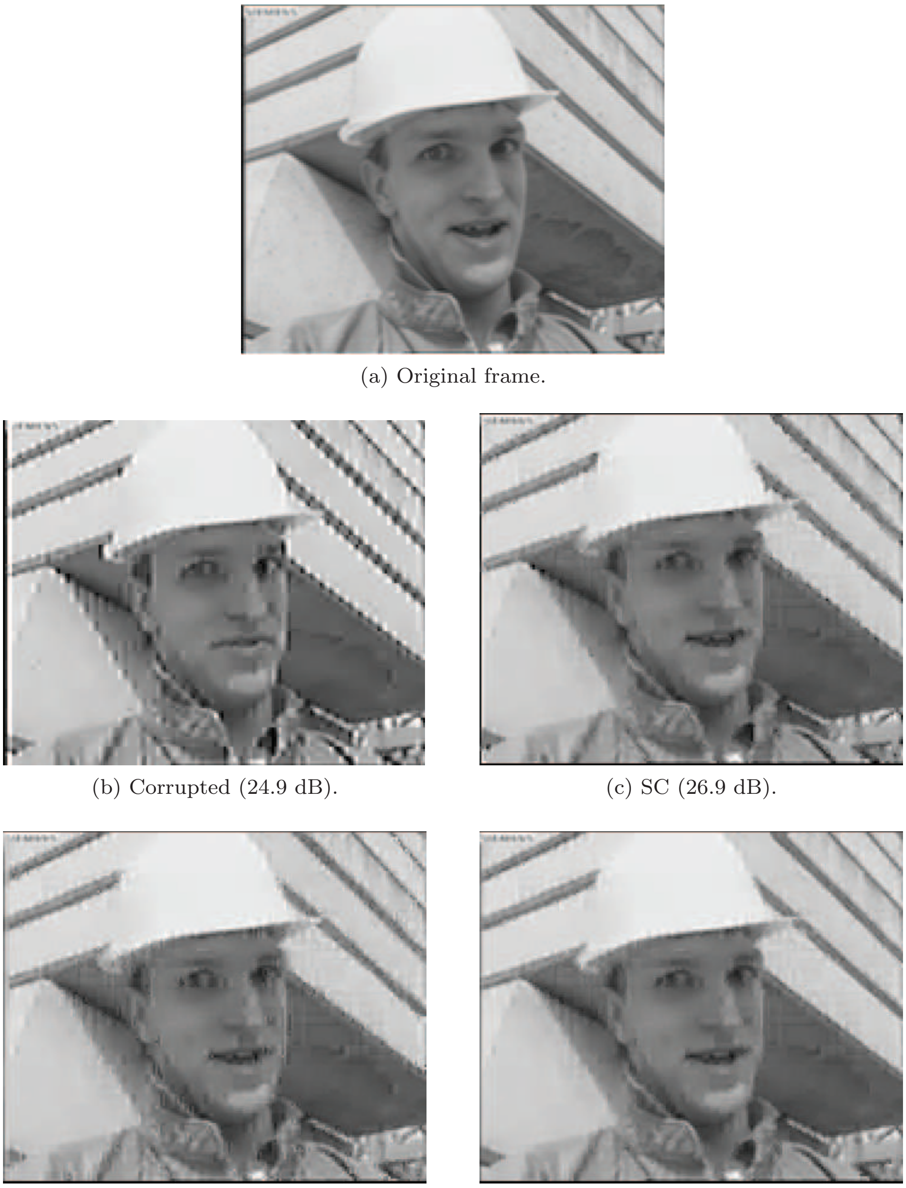


Figure 4.19: PSNR of concealed WZ frames comparing the different DVC EC methods.



(a) Original frame.

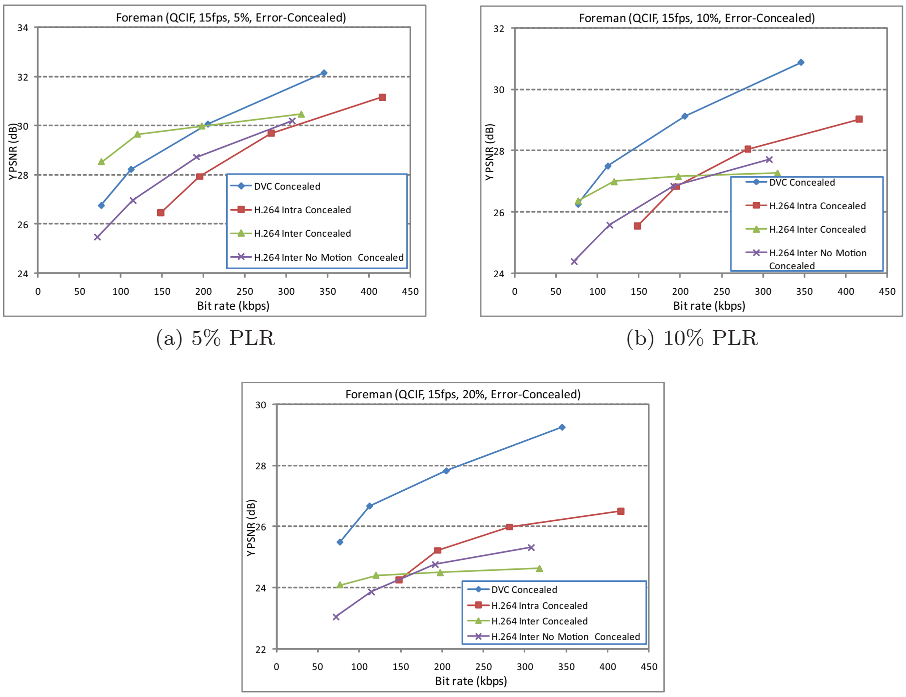
(b) Corrupted (24.9 dB).

(c) SC (26.9 dB).

(d) TC (28.9 dB).

(e) Proposed EC (31.6 dB).

Figure 4.20: Error-Concealed frame in *Foreman*.

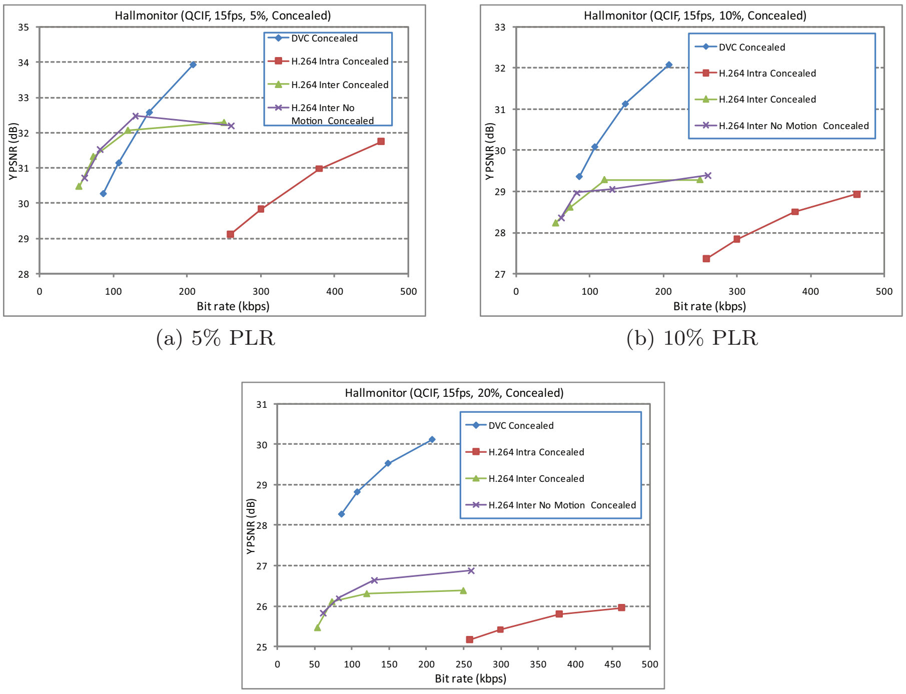


(a) 5% PLR

(b) 10% PLR

(c) 20% PLR

Figure 4.21: Error Resilience Performance of *Foreman* comparing the hybrid EC for DVC and EC for AVC/H.264.



(a) 5% PLR

(b) 10% PLR

(c) 20% PLR

Figure 4.22: Error Resilience Performance of *Hallmonitor* comparing the hybrid EC for DVC and EC for AVC/H.264.

low PLR (Figure 4.21(a)), AVC/H.264 (Intra, Inter, and Inter no motion) can compensate the channel errors and for higher PLR values (Figure 4.21(b) and 4.21(c)), AVC/H.264 Inter encoding gives the worst results, especially at high bit rates. On the contrary, the EC for DVC significantly improves the quality for all PLRs at high bit rates. Furthermore, it is worth noting that the key frames of DVC in our simulations are encoded with AVC/H.264 Intra mode, which means the error resilience performance of DVC is affected by that of AVC/H.264 Intra.

Similar results have been obtained for sequences with low motion such as *Hall-monitor*, as shown in Figure 4.22. The performance of AVC/H.264 Inter and H.264/AVC Inter no motion for such low motion sequence are very close. These results support similar conclusions from *Foreman* that the distortions caused by transmission errors in DVC are much smaller than those in AVC/H.264.

4.6 Conclusion

A new side information generation scheme for enhanced reconstruction is proposed to improve the performance of DVC in this chapter. The use of the partially decoded WZ frame, which is exploited in motion vector refinement and smoothing, and optimal compensation mode selection, improves the performance of SI generation. Simulation results show significant gain in RD performance over the TDWZ are achieved for all test sequences, especially the ones with large motion. These results are obtained with no additional complexity or modification to the DVC encoder.

The new SI generation scheme also has a significant contribution to the proposed hybrid EC scheme for WZ frames. The error-concealed frame by spatial EC based on an edge directional filter is used as a partially error-concealed frame. The latter is used to improve the performance of the temporal EC. Simulation results on the proposed EC method also show significant improvements in terms of both objective and perceptual qualities for the corrupted sequences. The proposed hybrid EC scheme in only applied at the decoder, without any modification to the DVC encoder or transmission channels. Furthermore, the hybrid EC of DVC is compared with error-concealed AVC/H.264. After concealment, the simulation results show that the distortions caused by transmission errors in DVC are much smaller than those in H.264/AVC. Moreover, the DVC codec significantly outperforms AVC/H.264 in its different modes for high PLR and high bitrates,. Finally, The results also confirm the intrinsic error resilience capability of DVC as it is based on a statistical approach rather a deterministic one.

Chapter 5

Multiview Distributed Video Coding

5.1 Introduction

Multiview video is attractive for a wide range of applications such as Free Viewpoint Television (FTV) [98] and video surveillance camera networks. The increased use of multiview video systems is mainly due to the improvements in video technology. In addition, the reduced cost of cameras encourages the deployment of multiview video systems.

FTV is one of the promising applications of multiview. FTV is a 3D multiview system that allows viewing the scene from a view point chosen by the viewer. Video surveillance is another area where multiview can be beneficial for monitoring purposes. In addition, the multiple views can be used to improve the performance of event detection and recognition algorithms. However, the amount of data generated by multiview systems increases rapidly with the number of cameras. This makes data compression a key issue in such systems.

In this chapter, a review of different SI techniques for multiview DVC is first provided, including a through evaluation of their prediction quality, complexity and RD performance. Moreover, all the SI techniques are combined in the Ground Truth (GT) fusion, which combines the different side informations using the original WZ frame at the decoder. Even though this is not feasible in practice, it gives the maximum achievable DVC performance.

Further, a new technique called Iterative Multiview Side Information (IMSI) is proposed to improve the DVC RD performance especially for video with significant motion. IMSI uses an initial SI to decode the WZ frame and then constructs a final SI which is used in a second reconstruction iteration.

At the same time, the GT fusion shows how the different side informations are correlated. Therefore, fusion algorithms between the least correlated side informations are introduced as they would represent a good trade-off between performance improvement and complexity increase. The first fusion algorithm is completely performed at the decoder side as it uses the key frames as estimates of the WZ frame to perform the fusion on a pixel basis. The second algorithm requires a binary mask computed at the encoder in addition to the key frames to perform the fusion. This mask informs the decoder of the decision pixel to

use, previous or forward key frame pixel, in the fusion.
Finally, the performance of multiview DVC is compared with respect to AVC/H.264 [13] Intra, No Motion (i.e. zero motion vectors) and Inter Motion.
The remaining of this chapter is structured as follows. First, Multiview DVC is described in section 5.2, whereas, section 5.3 reviews the different inter-camera prediction techniques. The IMSI technique is proposed in section 5.4. Then, the test material and simulation results are presented and discussed in section 5.5. Finally, some concluding remarks are drawn in section 5.6.

5.2 Multiview DVC (MDVC)

MDVC is a solution that allows independent encoding of the cameras and joint decoding of the different video streams as shown in Figure 5.1.

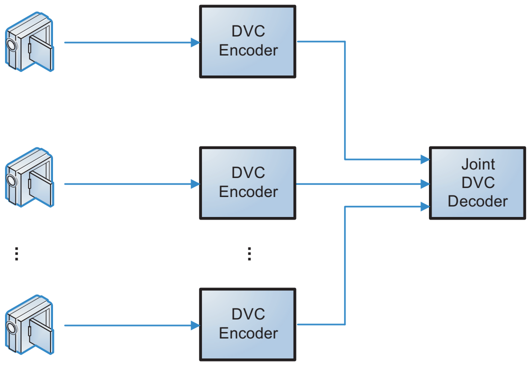
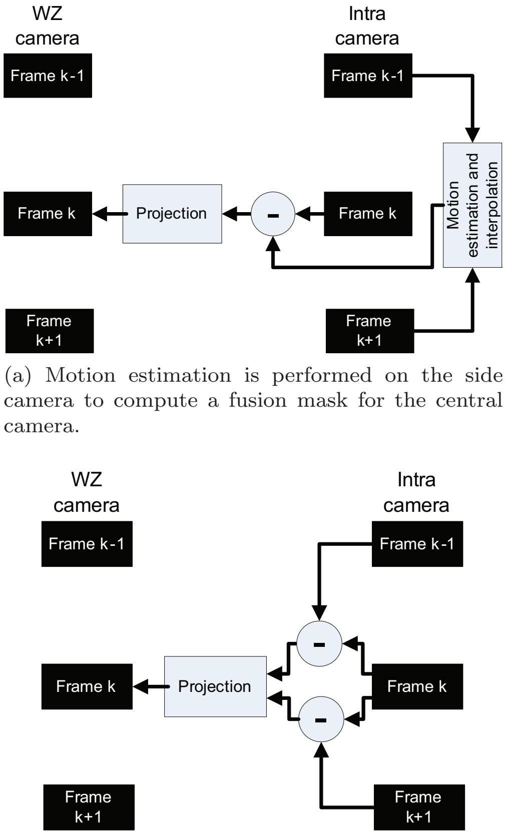


Figure 5.1: MDVC scheme. The different views are separately encoded and jointly decoded.

It differs from monoview DVC in the decoder. More precisely, the SI is constructed not only using the frames within the same camera but using frames from the other cameras as well.
Artigas et al. [99] proposed two novel fusion techniques between temporal and inter-camera side informations. In the first technique, temporal motion interpolation is performed between the previous and the forward frames from the side cameras. The result is subtracted from the current frame and then thresholded to obtain a binary mask. The latter is projected to the central camera to perform the fusion as shown in Figure 5.2 a). The second algorithm uses the previous and the forward frames as predictors for the current frame on the side cameras to compute a reliability mask. The latter is projected to the central camera and used to perform the fusion as depicted in Figure 5.2 b). It is reported that the fusions improve the average PSNR of the SI using high resolution video (1024x768). On the other hand, the RD performance of DVC is not investigated and the simulations are run using the originals, which is in practice not feasible. Moreover, depth maps are required to perform the inter-camera prediction, which is a hard problem for complex real world scenes.
In [100], the wavelet transform is combined with turbo codes to encode a multiview camera array in a distributed way. At the decoder, a fusion technique is introduced to combine temporal and homography-based side informations.



(a) Motion estimation is performed on the side camera to compute a fusion mask for the central camera.

(b) Frame difference w.r.t the previous and forward frames on the side camera is used to compute the fusion mask.

Figure 5.2: Fusion techniques proposed by Artigas et al. [99].

It thresholds the motion vectors and the difference between the corresponding backward and forward predictions to obtain a fusion mask. The mask assigns the regions with significant motion vector and prediction error to homography SI and the rest is assigned to temporal SI (i.e. regions with low motion and relatively small error prediction). It is reported that the hybrid SI outperforms the temporal one by around 1.5 dB in PSNR. In addition, it outperforms H.263+ Intra by around 4.0∼7.0 dB.

Further, a flexible prediction technique that can jointly utilize temporal and view correlations to generate side information is proposed in [101]. More specifically, the current pixel in the WZ frame is mapped using homography to the left and right camera frames. Then, AVC/H.264 decision modes are applied to the pixel blocks in the left and right camera frames. If both resulting modes are inter modes, the SI value is taken from temporal SI. Otherwise, it is taken from homography SI. The simulation results show that this technique significantly outperforms conventional H.263+ Intra coding. Nevertheless, comparison with AVC/H.264 Intra would be beneficial as it represents state of art for conventional coding.

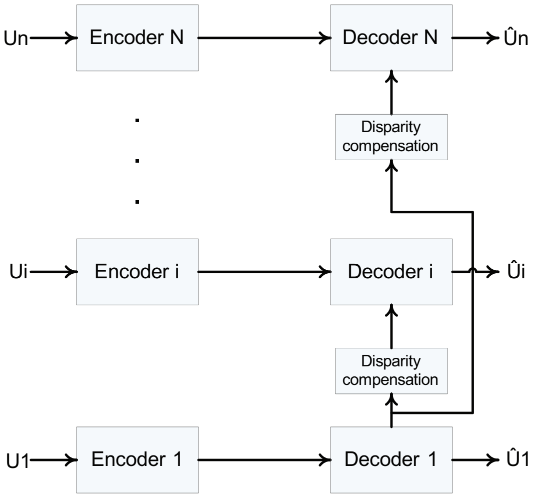


Figure 5.3: Distributed coding scheme with disparity compensation at the central decoder [102].

In [102], coding of multiview image sequences with video sensors connected to a central decoder is investigated. The N sensors are organized in an array to monitor the same scene from different views as shown in Figure 5.3. Only decoders 2 to N perform DVC using disparity compensated output of decoder 1. In addition, the video sensors are able to exploit temporal correlation using a motion compensated lifted wavelet transform [103] at the encoder. The proposed scheme reduces the bit rate by around 10% by performing joint decoding when compared to separate decoding.

Finally, ways of improving the performance of multiview DVC are explored in [104]. Several modes to generate homography-based SI are introduced. The homography is estimated using a global motion estimation technique. The results show an improvement of SI quality by around 6.0 dB and a gain in RD performance by around 1.0∼2.0 dB. However, the reported results assume an ideal fusion mask, which requires the knowledge of the original at the decoder. This is not feasible in a practical scenario.

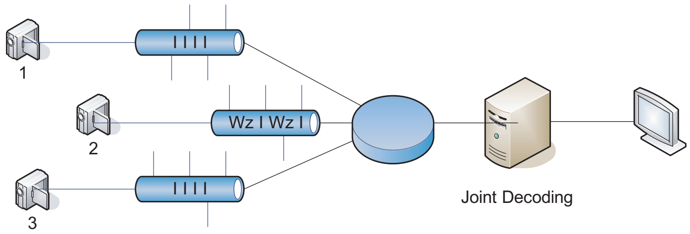


Figure 5.4: The multiview camera setup considered in this work. I stands for Intra frame and WZ for Wyner-Ziv frame.

5.3 Inter-Camera Prediction

In this section, different SI techniques for multiview DVC are reviewed. The different techniques are described for a 3 camera setup, where the central camera is predicted from both neighboring cameras, as depicted in Figure 5.4.

5.3.1 Disparity Compensation View Prediction (DCVP)

DCVP [105] is based on the same idea as MCTI, but the motion compensation is performed between the frames from the side cameras. A slight modification is applied to DCVP to improve the SI quality. Instead of interpolating the motion vectors at midpoint, an optimal weight is computed in [105]. For this purpose, the first frame of each camera is conventionally decoded. Then, motion compensation is performed between the side camera frames. The motion vectors are weighted with the weights 0.1,0.2, ... 0.9. Further, the SI PSNR is computed for each weight. The weight with maximum PSNR is maintained and used for the rest of the sequence. Nevertheless, the SI generated by DCVP has usually a poorer quality than the one generated by MCTI. This is due to the larger disparity between the side camera frames when compared to the one between the previous and forward frames.

5.3.2 Homography

The homography, H, is a 3x3 matrix transforming one view camera plane to another one as shown in Figure 5.5.
It uses eight parameters a, b, c, d, e, f, g and h. The homography maps a point (x_1,y_1) from one plane to a point (x_2,y_2) in the second plane up to a scale λ such that

$$\lambda \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}. \tag{5.1}$$

This model is suitable when the scene can be approximated by a planar surface, or when the scene is static and the camera motion is a pure rotation around its

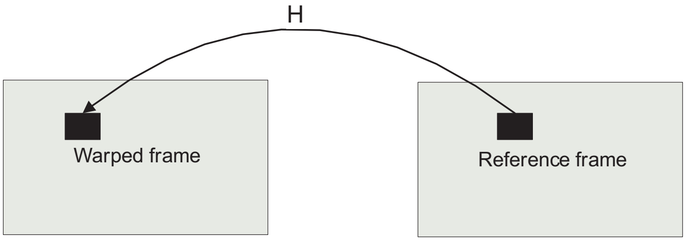


Figure 5.5: Homography matrix H relating one view to another.

optical center. The homography can be calculated using various techniques. In this work, we consider a global motion estimation technique introduced in [106] to compute the homography. The parameters are calculated such that the sum of squared differences E between the reference frame and the warped side frame is minimized.

$$E = \sum_{i=1}^{N} {e_i}^2 \ with \ e_i = I_w(x_{w_i}, y_{w_i}) - I(x_i, y_i), \tag{5.2}$$

where I$_w(x_{w_i}, y_{w_i})$ and I(x_i, y_i) are the pixels from the warped and reference frames, respectively. The problem is solved using the Levenberg-Marquardt gradient descent algorithm to iteratively estimate the parameters. To remove the influence of such outliers, a truncated quadratic is used. In other words, only pixels for which the absolute value of the error term is below a certain threshold are taken into account in the estimation process, other pixels are ignored. Therefore, the algorithm will count mainly for global motion.

$$E = \sum_{i=1}^{N} \rho(e_i) \ with \ \rho(e_i) = e_i^2 \ if|e_i| \geq T \ else \ 0, . \tag{5.3}$$

where T is a threshold.
In multiview DVC, the warped frame is computed from the left (H_L) and right (H_R) camera frames as shown in Figure 5.6. Therefore, three side informations are possible. The one entirely warped form each side camera and the average (H) of both side cameras. The latter is the only one considered in this work.

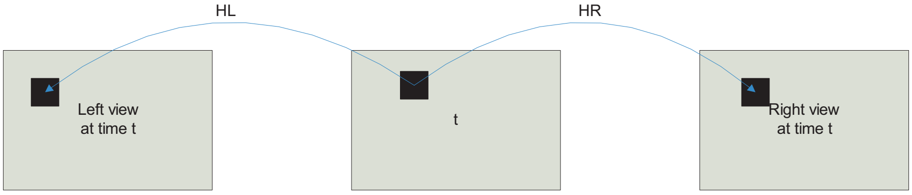


Figure 5.6: Homography-based SI.

The advantage of this technique is that once the homography relating the central camera with the side ones is estimated, computing the SI becomes very simple in terms of computational complexity when compared to techniques based on exhaustive block-based motion estimation. Moreover, this technique is suitable for scenarios where the global motion is highly dominant with respect to local variations as it would generate a good estimation in this case. On the other hand, if the scene has multiple significant objects moving in different directions, the estimation would be of a poor quality as the technique would only account for global motion.

5.3.3 View Synthesis Prediction (VSP)

The previously mentioned techniques do not take advantage of some important feature of multiview. That is the speed at which an object is moving in a view depends on its depth information. In addition to this, rotations, zooms and different intrinsic parameters are difficult to model using a motion vector, which is a simple translational model. Furthermore, the homography tries to estimate a global motion and ignores local motion using a truncated error function, which is not the case of VSP [107]. In the latter, the camera parameters, intrinsic and extrinsic, are used to predict one camera view from its neighbors.

For simplicity, the case of one neighboring camera is considered as shown in Figure 5.7. The view from camera c_2 can be synthesized from camera c_1. Each pixel $I(c_1,x,y)$ from camera c_1 is projected into the 3D world reference using its depth information.

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = A \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X_{3D} \\ Y_{3D} \\ Z_{3D} \\ 1 \end{pmatrix}, \tag{5.4}$$

where A is the intrinsic parameters matrix, and R and T are the rotation and translation matrices with respect to the 3D world reference. Moreover, the depth information is equal to Z_{3D}, which corresponds to the Z coordinate of the point in the 3D world coordinates. It is substituted in equation 5.4 and the resulting system is solved for X_{3D} and Y_{3D}. Then, the 3D point is projected back to the 2D plane of camera c_2. This process is performed for each pixel of camera c_1.

In the multiview camera setup used in this research, the pixel in the central camera is mapped to both side cameras. The pixel value is taken as average of both side camera pixels.

The drawback of this technique is the difficulty to estimate depth for real world complex scenes. In addition, the quality of the SI depends on the precision of the camera calibration and depth estimation.

5.3.4 View Morphing (VM)

Image morphing can generate compelling 2D transitions between images. However, differences in object pose or viewpoint often cause unnatural distortions in image morphs. Using basic principles of projective geometry, one can perform a simple extension to image morphing that correctly handles 3D projective camera and scene transformations. The view morphing requires the computation of the fundamental matrix, which is the algebraic representation of epipolar geometry. Suppose that we have a point P in the 3D world coordinates. This point is visible in both cameras with optical centers C_0 and C_1 as P_0 and P_1

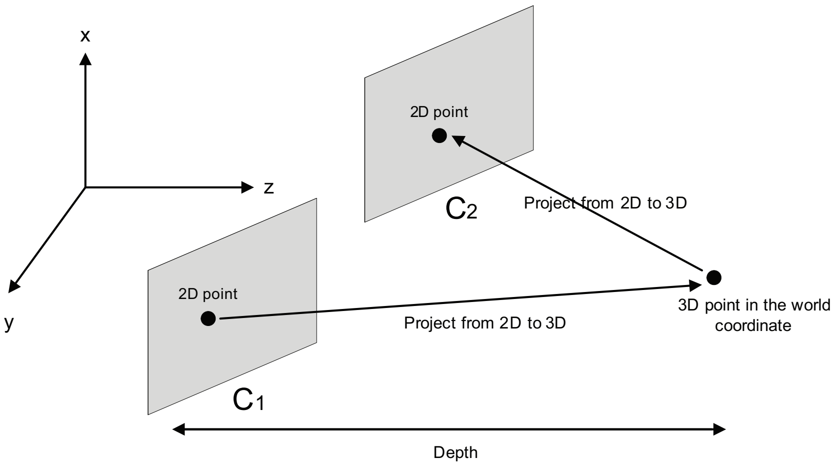


Figure 5.7: View Synthesis Prediction.

respectively. The three points P, C_0 and C_1 define a plane called the epipolar plane π. The line intersection of the epipolar plane with each image plane is called an epipolar line as shown in Figure 5.8. The fundamental matrix is derived from the mapping between a point in one camera and its epipolar line in the other camera. Therefore, matching points should be calculated between the two images.

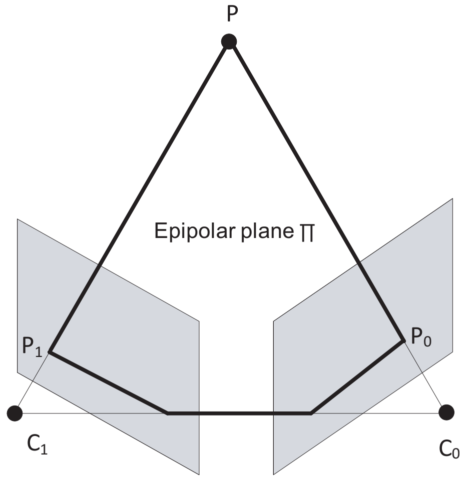


Figure 5.8: The epipolar line and plane.

VM [108] is used to get an image from a virtual camera that could be placed between two real cameras as shown in Figure 5.9. The input of the view morphing algorithm is two images from real cameras and information about the correspondences between regions in the two images or projection matrices of the side cameras from 3D world coordinates to 2D coordinates in each camera's plane. The output of the algorithm is a synthesized image (i.e. a view from the virtual camera).

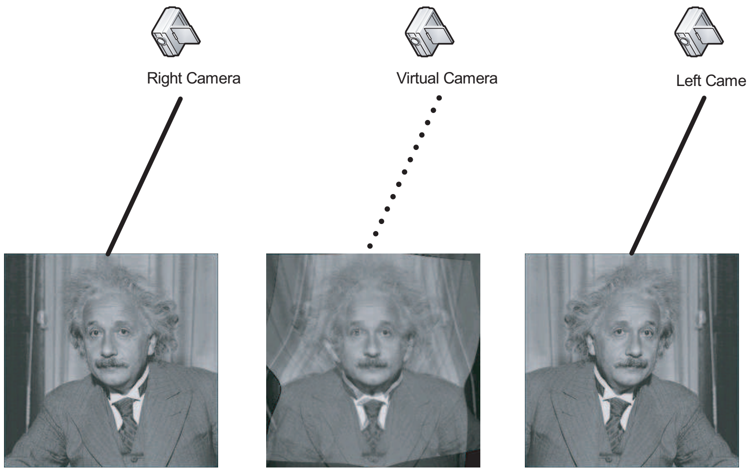


Figure 5.9: The virtual camera in view morphing.

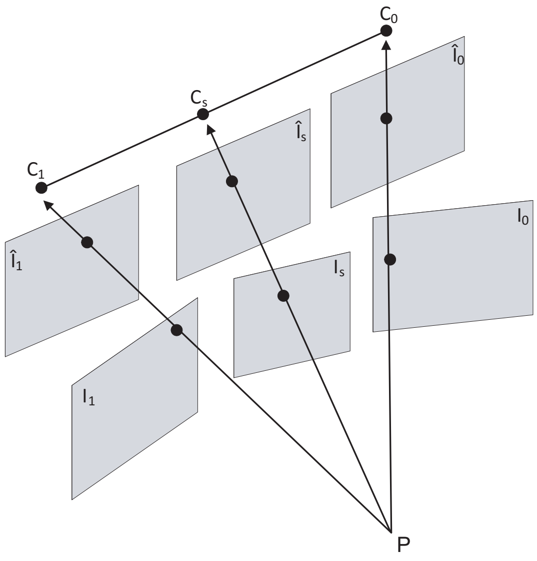


Figure 5.10: The view morphing algorithm.

The morphing algorithm is illustrated in Figure 5.10. Initially, both images I_0 and I_1 are warped across the scanlines to get $\hat{I}_0$ and $\hat{I}_1$ respectively, which are in the same plane. The latter are morphed across the position of the virtual camera C_s to get $\hat{I}_s$. Finally, $\hat{I}_s$ is unwarped to get I_s. As in the case of DCVP, an optimal weight s is computed for the virtual camera C_s such that the PSNR is maximized for the warped frame with respect to the central view frame.

The problem with VM is that it works very well for simple scenes with a central object in front a uniform background. In this case, extracting matched feature points with a high degree of accuracy from the scene is simple as these points are used to compute the fundamental matrix. On the other had, VM fails for real world scenes as the matched feature points task becomes a more challenging task.

5.3.5 MultiView Motion Estimation (MVME)

MVME [109] finds the motion vectors in the side cameras and then applies them to the central camera to estimate the WZ frame as shown in Figure 5.11. The motion vectors computed in one view should be transformed before being used in another view. Nevertheless, they can be directly reused if all the cameras lie in the same plane and point in the same direction.

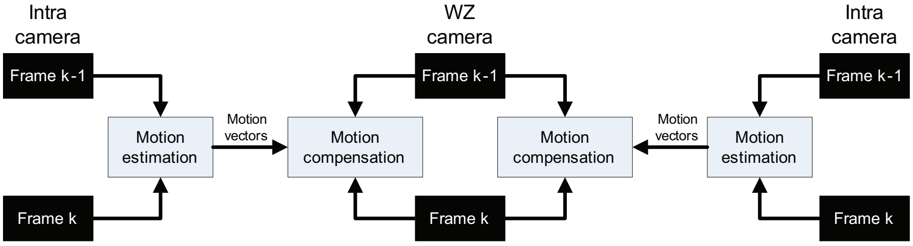


Figure 5.11: Conceptual scheme. Motion vectors are found in the intra camera and used in the WZ camera.

First, a *disparity vector* $\vec{dv}$ is obtained by block-based full search between the WZ and the Intra cameras for frame k-1. The vector $\vec{dv}$ estimates the location of each block from the WZ camera in the Intra camera. Then, the *motion vector* $\vec{mv}$ is computed by searching in frame k in the Intra camera for the best match for the block obtained in the previous step as illustrated in Figure 5.12-A. Finally, the *motion vector* $\vec{mv}$ is applied to the aligned block in frame k in the WZ camera as depicted in Figure 5.12-B.

Figure 5.13 shows the possible motion paths to estimate the WZ frame, which are a total of 8 paths, 4 inner paths and 4 outer paths, each generating one estimate. The inner paths are computed as described above by performing disparity estimation followed by motion estimation on the Intra camera (Figure 5.13 a)). The outer paths are computed by doing the opposite of inner paths computation, starting with motion estimation on the Intra camera followed by disparity estimation (Figure 5.13 b)). The simplest way to generate the final SI is by taking the average of these estimates. A better strategy is to compute a reliability measure for each path on a block or pixel basis and weight the estimates before taking the sum. For this purpose, Mean Square Error (MSE)

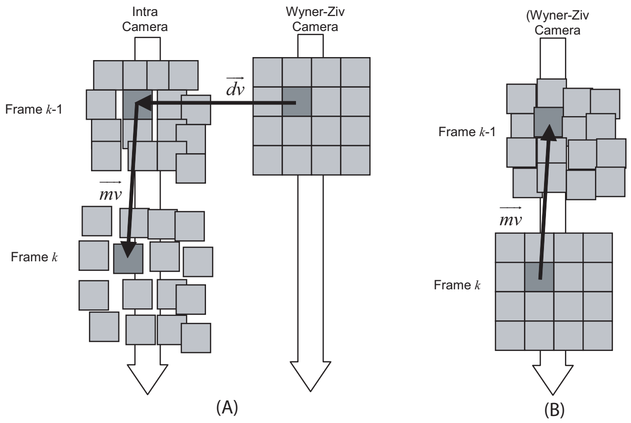
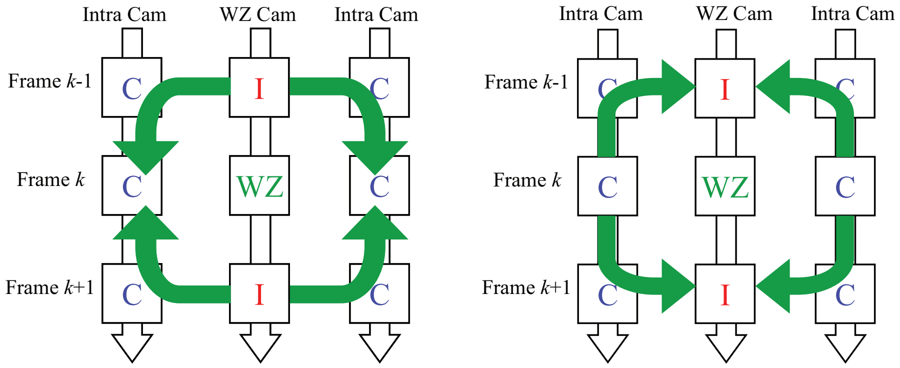


Figure 5.12: A)Motion estimation scheme. B)Motion compensation scheme [109].



(a) 4 Inner paths.

(b) 4 Outer paths.

Figure 5.13: The 8 possible paths when using two intra cameras and two reference frames in each camera [109].

or Mean Absolute Difference (MAD) computed between the original and the candidate blocks is used as a reliability measure.

5.4 Iterative Multiview Side Information (IMSI)

We initially introduced iterative SI for the monoview scenario in [110], where the final SI depends not only on the key frames but also on the WZ bits as well. This final SI is used to refine the reconstruction of the decoded WZ frame. This is done by running the reconstruction process in a second iteration to enhance the quality of the decode frame. The process of IMSI is illustrated in Figure 5.14.

Initially, the reconstruction process of DVC is described in this section. Then, IMSI is introduced.

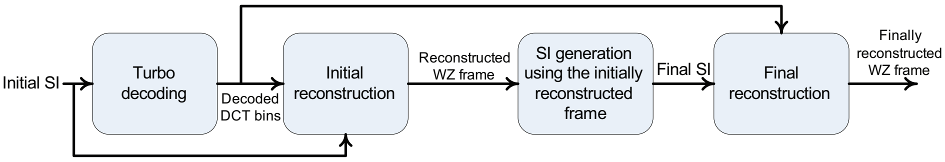


Figure 5.14: The IMSI generation process.

5.4.1 DVC Reconstruction

This stage in the decoding process is opposite to the quantization step at the encoder. After turbo decoding, the reconstruction block uses the SI along with decoded WZ DCT bins to improve the reconstruction quality as described in [9]. The principal consists in either accepting a SI value as a reconstructed value if it fits into the quantization interval corresponding to the decoded bin or truncating the SI value into this quantization interval.

Let Y be the SI value, d the decoded quantized index, Δ the quantization step and $\hat{X}$ the reconstructed value. In the case of the DC band, the reconstructed value $\hat{X}$ is computed as

$$\hat{X} = \begin{cases} Y & \text{if } d\Delta \leq Y \leq (d+1)\Delta \\ d\Delta & \text{if } Y < d\Delta \\ (d+1)\Delta & \text{if } Y > (d+1)\Delta \end{cases}$$

For the AC bands, the reconstructed value $\hat{X}$ is computed in a similar way.

5.4.2 IMSI for Enhanced Reconstruction

First, the initial SI to use in the WZ frame decoding is chosen depending on the nature of the video. This is done by computing the average luma variation per pixel between the key frames at the decoder, which is compared to a threshold. If it is below the threshold, the motion is considered not significant and MCTI is used as the initial SI. Otherwise, MVME is taken as initial SI. This is motivated by the results presented further in section 5.5.2. Namely, MCTI shows better prediction quality for low motion video content. On the other hand, MVME is shown to have a better performance for video with significant motion.

Then, WZ decoding is performed using the initial SI, which implies turbo decoding followed by a reconstruction stage. Then, the decoded WZ frame from the first stage is predicted by block-based motion search and compensation as in conventional video coding using four references: the previous, forward, left camera and right camera frames. More specifically, for each block in the decoded frame, the best matching block with minimum distortion is selected using the SAD (Square Absolute Difference) as the distortion metric as shown in Figure 5.15. This generates a final SI. It is important to stress the fact that this method does not use the original WZ but rather the decoded WZ frame using the initial SI.

Finally, the final SI is used in a second iteration in the reconstruction block.

IMSI is expected to be efficient in situations where motion is significant as the difference in prediction quality between the initial and final SI is more important. The reason is that the final SI is highly correlated with the WZ frame in the case of high activity video content. Therefore, most of the SI values map into the decoded bin in the reconstruction process (i.e. the SI value is taken as the reconstructed value). This produces a better reconstruction with lower distortion as less SI values are truncated into the quantization interval, when compared to the initial reconstruction phase, using the initial SI.

The improvement for low motion video is negligible as both side informations, initial and final, are close in terms of prediction quality.

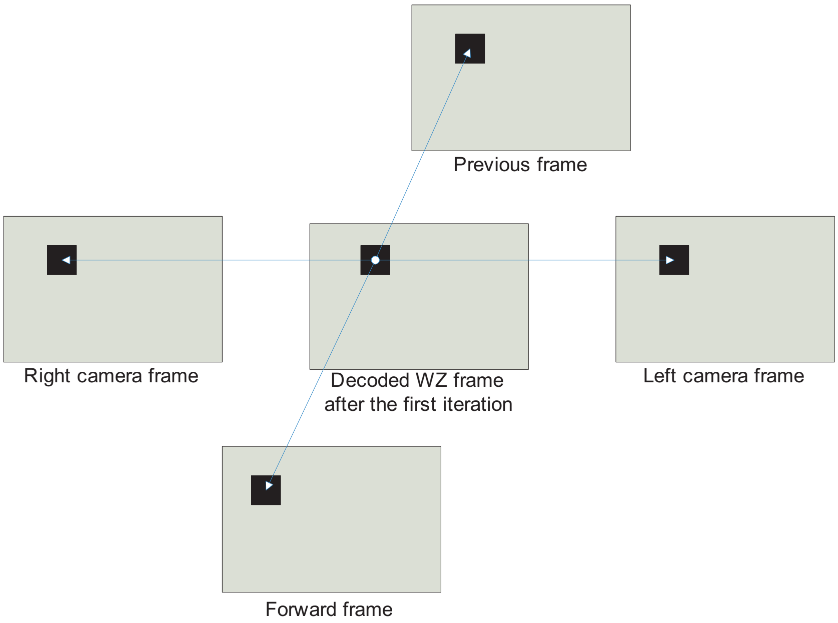


Figure 5.15: The final SI construction in IMSI

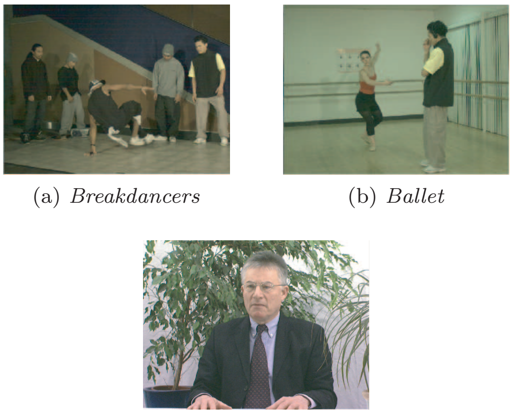
IMSI generates a very good estimation of the WZ frame since it uses the decoded WZ frame from the first iteration to compute the prediction. On the other hand, the price to pay for this good estimation is the initial WZ rate spent to initially decode the WZ frame. In addition, the decoder's complexity significantly increases due to the additional motion search task.

5.5 Simulation Results

5.5.1 Test Material and Evaluation Methodology

The sequences *Breakdancers*, *Ballet* and *Uli* shown in Figure 5.16 are used for evaluating the performance of the different SI techniques. *Breakdancers* and *Ballet* contain significant motion. This makes the motion estimation a difficult and challenging task. On the other hand, *Uli* is a conference-like video sequence, which contains more or less static video content. The spatial resolution is 256x192 for all the sequences. The temporal resolutions are 15 fps for *Breakdancers* and *Ballet*, and 25 fps for *Uli*.

In this research, three camera views are used and the performance is evaluated



(a) *Breakdancers* (b) *Ballet*

(c) *Uli*

Figure 5.16: Sequences *Breakdancers*, *Ballet* and *Uli*.

only for the central camera. For DVC simulations, the DISCOVER codec [17] is run with the following settings:

- Only luminance data is coded.
- The central camera is the only one containing WZ frames. The side cameras (i.e. left and right) are conventionally encoded in the Intra mode while the central one contains WZ frames, as depicted in Figure 5.4.
- The same Quantization Parameter (QP) is used for the side cameras and the key frames of the central camera. A QP is defined per quantization matrix such that the decoded key and WZ frames have a similar quality.
- The GOP size is equal to 2.
- Four RD points are computed per SI.

For AVC/H.264 coding, the publicly available reference software (JM 11.0) [76] is used with the following settings:

- Intra, No Motion and Inter Motion modes. For the No Motion mode, each motion vector is equal to zero, which means that each block in a P frame is predicted from the co-located block in the previous I frame. For the Inter Motion mode, the motion search range is set to 32. In both modes, the GOP size is equal to 12.
- High profile with CABAC.
- The 8x8 transform enabled.

5.5.2 Side Information Quality

In this section, the SI PSNR is evaluated for the SI techniques at the different RD points. *Uli* is not provided with depth maps. In addition, the feature point matching performs poorly due to highly textured scene background in the sequence. For this reason, the VSP and VM techniques are not evaluated for *Uli*.

For IMSI, Figure 5.17 shows the luma pixel variation between the key frames for the three video sequences at the highest RD point. By picking a threshold equal to 1.7, *Breakdancers* and *Ballet* are classified as sequences with significant motion (i.e. MVME is used as the initial SI) and *Uli* is classified as a low motion video content (i.e. MCTI is used as the initial SI) at all RD points.

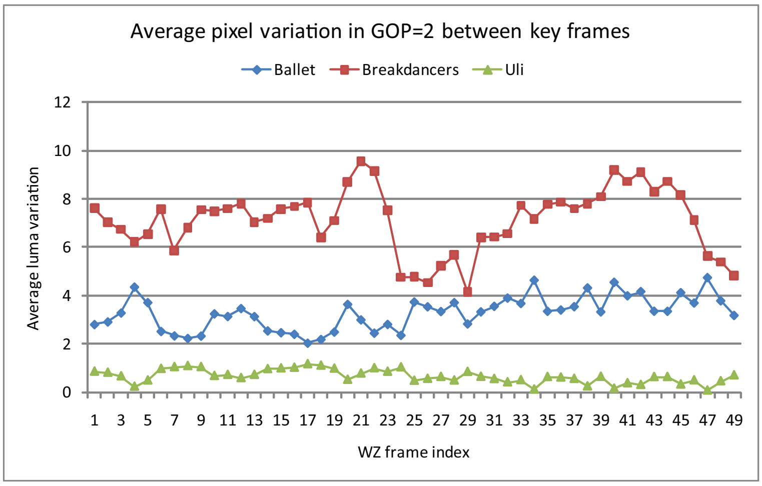


Figure 5.17: Average luma pixel variation for *Breakdancers*, *Ballet* and *Uli* at the highest RD point.

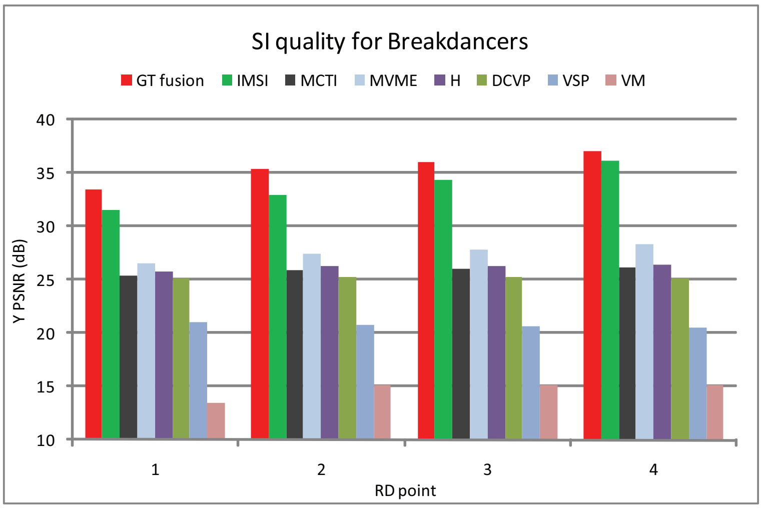


Figure 5.18: Side information quality for *Breakdancers*.

Figures 5.18, 5.19 and 5.20 show the SI PSNR for *Breakdancers*, *Ballet* and *Uli* respectively. Obviously, the GT fusion and IMSI produce the best prediction for all sequences at all RD points as they use respectively the original frame and the decoded WZ frame to construct the prediction. Thus, the comparison will mainly focus on the other SI techniques.

For *Breakdancers*, MVME produces the best SI quality followed by MCTI. On the other hand, the worst performance is for VSP. However, VSP requires two input parameters, camera calibration and depth estimation. The quality of the SI depends on the precision of these parameters. We can observe that most of the techniques perform quit well in terms of SI quality for this sequence as homography and DCVP are quite close to MCTI in prediction quality.

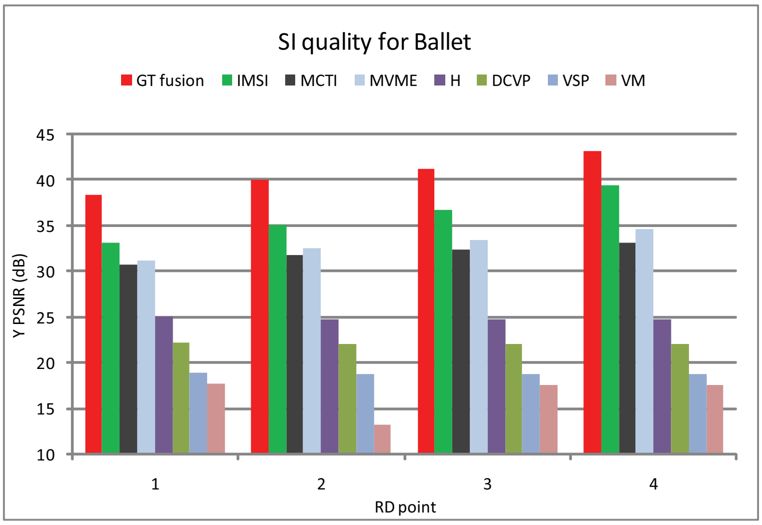


Figure 5.19: Side information quality for *Ballet*.

For *Ballet*, MVME produces the best SI quality followed by MCTI. *Ballet* contains motion but it is less significant than in the *Breakdancers* case. This explains the increase in PSNR gap between MCTI and the other SI techniques. As for *Breakdancers*, we have homography followed by DCVP, then VM and finally VSP in a decreasing order in terms of SI quality.

Since *Uli* contains little motion, we expect MCTI and MVME to work very well since MCTI performs a pure temporal interpolation and MVME performs an inter camera disparity estimation followed by a temporal motion estimation.

In summary, we can see clearly that MVME and MCTI produce by far better predictions than other SI generation techniques for *Ballet* and *Uli*. On the other hand, MVME, MCTI, Homography and DCVP are not very far from each other in terms of SI quality for *Breakdancers*.

Figure 5.21 illustrates the contribution of the different side informations to the GT fusion for *Breakdancers*. It is obvious that MCTI has the largest contribution around 43%∼55% out of the total number of frame pixels. It is followed by homography-based SI. The homography is the one that brings most innovation to the GT fusion. MVME and DCVP are highly correlated with MCTI. This is explained by the fact that these methods are of the same block-based nature. Finally, VSP and VM have the worst contribution to the GT fusion.

The contribution of the different side informations to the GT fusion for *Ballet* is illustrated in Figure 5.22. As for *Breakdancers*, MCTI has the largest contribution around 45%∼64%. It is larger than in the *Breakdancers* case since *Ballet* contains less motion than *Breakdancers*. It is followed by homography-based SI.

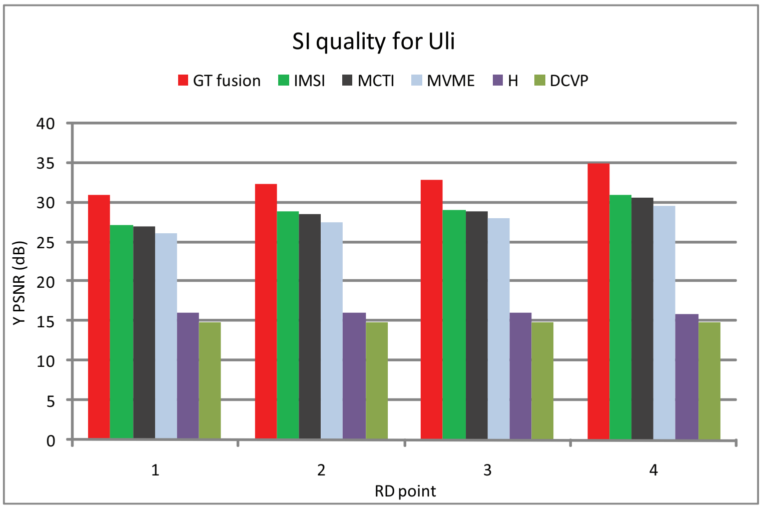


Figure 5.20: Side information quality for *Uli*.

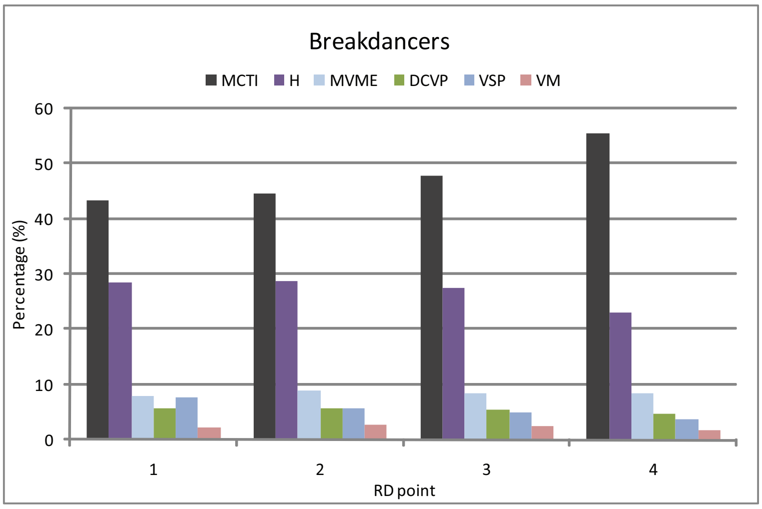


Figure 5.21: The percentage of contribution of the different side informations in the GT fusion for *Breakdancers*.

Then, MVME comes in third place followed by DCVP. Finally, VSP and VM are the worst in terms of contribution to the GT fusion.

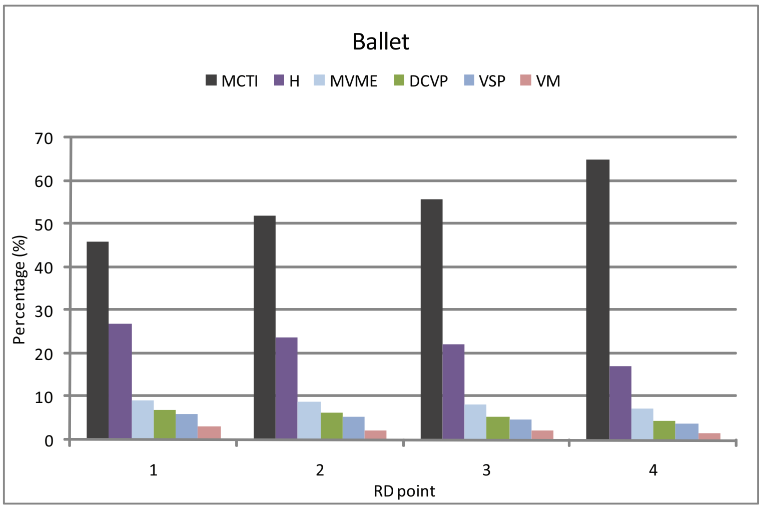


Figure 5.22: The percentage of contribution of the different side informations in the GT fusion for *Ballet*.

Since *Uli* contains low motion content, MCTI has the largest contribution to the GT fusion, around 54%∼73% out of all pixels. It is followed by homography-based SI and then MVME. Furthermore, the rest of side informations have a poor contribution to the GT fusion. This is illustrated in Figure 5.23.

For the three sequences, homography-based SI is the one that brings most innovations to the GT fusion as it is the least correlated SI with MCTI. Therefore, we can conclude that possible fusion algorithms combining MCTI and homography-based SI represent a good trade-off between performance improvement and complexity increase.

5.5.3 Side Information complexity

The different techniques complexities are compared in terms of the total number of arithmetic operations reuiqred to generate the side information. The image dimensions are H, the height, and W, the width. For the block based methods, a search range r and block size w are considered.

MCTI and DCVP

Both MCTI and DCVP have the same complexity. The only difference between both techniques is the input frames. For each block match, w^2 subtractions are required. Then, the error is computed, which requires $w^2 - 1$ additions. This is performed for each position within the search range. Thus, $(2w^2 - 1)r^2$ operations are required to find a match for each block. Finally, all the block should be processed. Therefore, $(2w^2 - 1) * r^2 * (H * W/w^2) \approx 2 * H * W * r^2$ is the number of operations required to estimate the motion between the two frames.

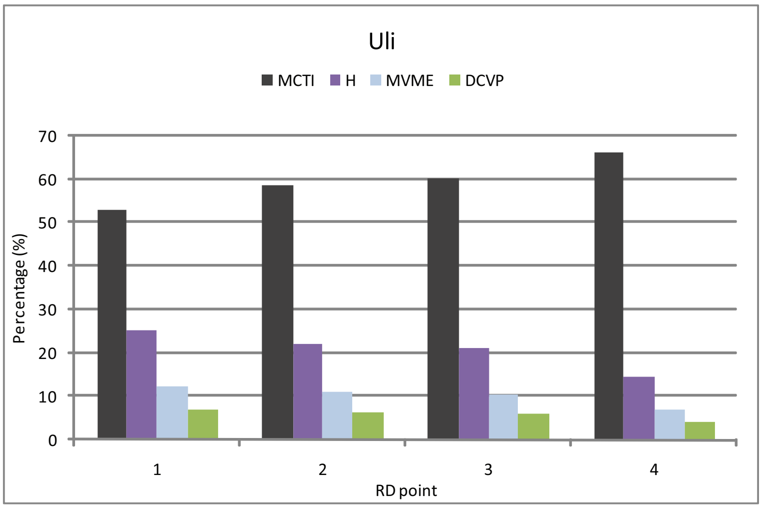


Figure 5.23: The percentage of contribution of the different side informations in the GT fusion for *Uli*.

MVME

There is a maximum of 8 paths. For each one, motion estimation is performed twice with the Intra camera and then across the side and the central cameras. Therefore, $2 * O(MCTI)$ operations are required for each path. Thus, a total of $16 * O(MCTI)$ operations is required for all the paths. In other words, MVME is approximately 16 times more complex than MCTI.

Homography

Initially, the homography matrices are computed offline. A total of 15 operations are required to compute the mapping for each pixel using the 3x3 homography matrix. Therefore, the complexity of the homography-based side information generation from both view is $2 * 15 * H * W = 30 * H * W$.

VM

In VM, both side frames are warped which requires $2 * 15 * H * W$ operations. Then, the resulting warped frames are morphed across the virtual camera position. The latter needs $3 * H * W$ operations. Finally, the morphed frame is unwarped to obtain the side information. Therefore, the total complexity is $3 * H * W + 3 * 15 * H * W = 48 * H * W$ operations.

VSP

For each pixel, the projection from the image plane to the 3D world coordinates requires 38 operations. Moreover, the projection back to the central camera requires 23 operations. This is performed for each pixel, which results in a total complexity of $61 * H * W$. It important to mention that this estimation does not take into account the depth estimation. This complexity applies given that the depth map is already available.

IMSI

The complexity of IMSI depends on the initial SI used, which is either MVME or MCTI. Then, the final SI generations requires $O(MCTI)$ operations. This implies a maximum complexity of $9 * O(MCTI)$ when MVME is used as the initial SI.

5.5.4 RD Performance

In this section, the RD plots for the different sequences are presented for the different side informations. It is important to mention that only SI with a significant RD performance are presented. Therefore, the performance of VM and VSP is not plotted for *Breakdancers* and *Ballet*. For *Uli*, only IMSI, MCTI and MVME are plotted as they significantly outperform the other side informations. On the other hand, the GT fusion combines all the side informations even the ones that are not plotted.

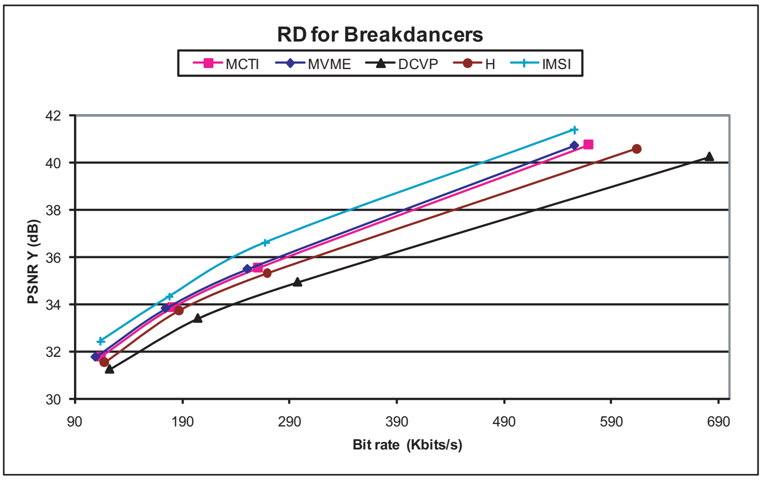


Figure 5.24: RD performance for *Breakdancers*.

For *Breakdancers*, IMSI has the best RD performance out of all SI techniques as it is superior to MVME by around 0.4 dB and 0.7 dB at low and high bit rates respectively. The SI quality is better for MVME than MCTI. This explains the performance gap between MVME and MCTI in Figure 5.24. This gap is more or less constant and around 0.2 dB. Further, homography and DCVP are inferior to MCTI by a maximum gap of around 1.0 dB and 2.0 dB respectively, at high bit rates. At average bit rates, this gap is around 0.5 dB and 1.2 dB respectively. The homography has a similar performance to MCTI at low bit rates and DCVP is inferior by 1.0 dB.

For IMSI, Figure 5.25 shows the quality of the reconstructed WZ frames for *Breakdancers* in the first and second reconstruction iterations for the highest RD point. In the initial one, around 13% of the SI values are truncated while this percentage is around 5% in the second reconstruction iteration resulting in a less distorted reconstruction.

For *Ballet*, IMSI has the best RD performance slightly outperforming MVME

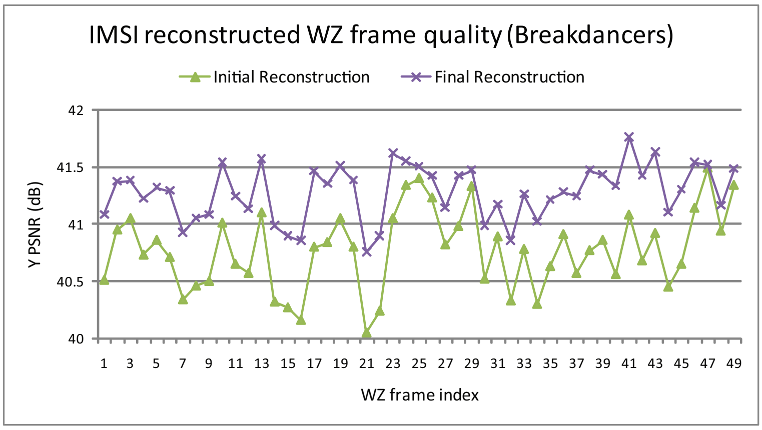


Figure 5.25: The reconstructed WZ frames quality for the initial and final reconstruction for *Breakdancers* for the highest RD point.

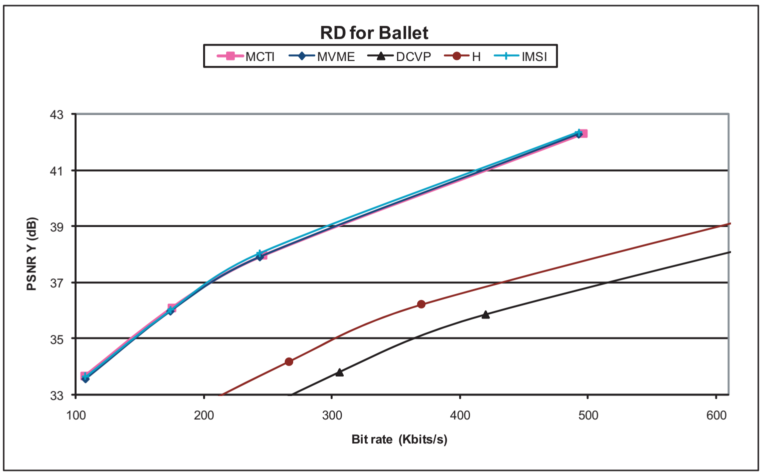


Figure 5.26: RD performance for *Ballet*.

by around 0.1 dB at high bit rates. Obviously, the performance improvement is less important than in the *Breakdancers* case as this sequence has less motion. Further, MVME and MCTI have a similar performance as shown in Figure 5.26. Even though MVME has a slightly better SI quality than MCTI for all RD points, it is not translated to a better RD performance. The reason is that the DVC scheme operates in the DCT domain not the pixel domain. Thus, a better SI PSNR, which is computed on the pixel values, does not automatically imply better performance for transform domain WZ decoding.

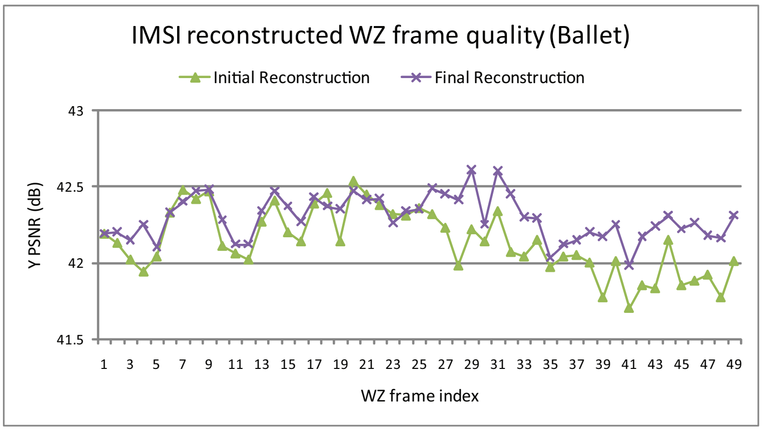


Figure 5.27: The reconstructed WZ frames quality for the initial and final reconstruction for *Ballet* for the highest RD point.

Finally, the reduction in the number of truncated SI values with IMSI is less significant (i.e. around 2%) for *Ballet* than in the case of *Breakdancers*. This leads to less improvement in the reconstruction as shown in Figure 5.27.

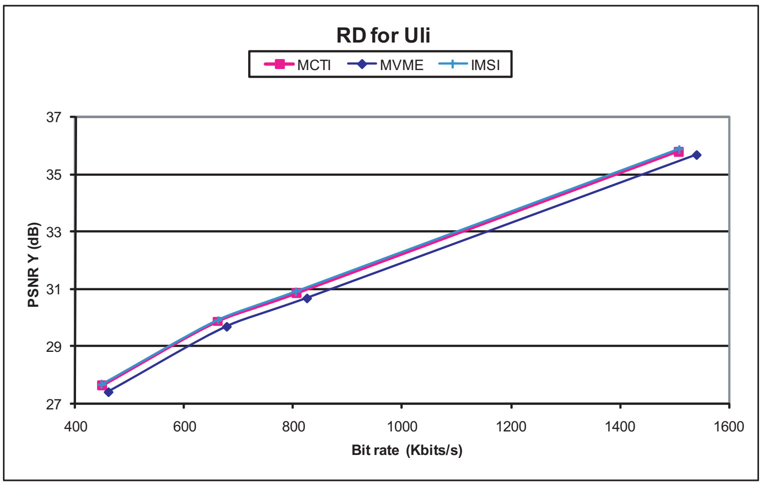


Figure 5.28: RD performance for *Uli*.

As mentioned previously, *Uli* contains very low motion video content due to its nature. Therefore, both IMSI and MCTI have the best performance, but IMSI does not bring any improvement in this case. Both side informations outperform MVME by around 0.5 dB as shown in Figure 5.28.

5.5.5 Fusion-based Side Information

By studying the correlation between the different side informations, MCTI and homography turned out to be the least correlated side informations. Thus, they would represent a good trade-off between performance improvement and complexity increase. Therefore, two fusion techniques between MCTI and homography are introduced in this section. The first one is completely preformed at the decoder while the second one is encoder driven as the encoder sends a binary mask to the decoder side to help in performing the fusion.

Decoder Driven Fusion

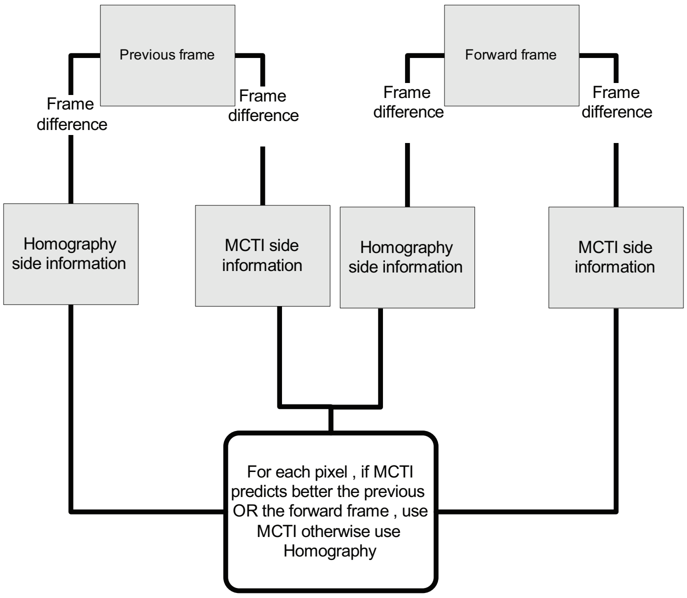

Previous frame
Forward frame
Frame difference
Frame difference
Frame difference
Frame difference
Homography side information
MCTI side information
Homography side information
MCTI side information
For each pixel , if MCTI predicts better the previous OR the forward frame , use MCTI otherwise use Homography

Figure 5.29: Fusion 1, the fusion mask is generated using the previous, forward and both side information frames.

In this case, the previous and the forward key frames are used as estimates of the WZ frame as the fusion mask is computed with respect to these two frames. More specifically, for each pixel from the previous frame, the pixel that predicts it better from both side informations is searched for. This is done by taking the difference between the current pixel from the previous frame and the pixel at the same position from both side informations. If the homography-based side information has a smaller error, then, the binary fusion mask is set to 1 at this position. Otherwise, if MCTI has a smaller error, the binary mask is set to 0. This process is illustrated in Figure 5.29. The same processing is applied to the forward frame. Thus, a second binary mask is obtained. Finally, a pixel-wise logic OR operation between both binary masks is performed to obtain the final fusion mask. The latter is used to construct the final side information such that

1 indicates that the pixel is taken from the homography-based side information. Otherwise, it is equal to 0, and the pixel is taken from MCTI. This algorithm is referred to as *decoder driven fusion* 1.

Next, a second decoder driven fusion algorithm based on motion vectors magnitude is described. since MCTI performs poorly in regions of the frames where motion is significant and the homography should work better in these regions. Therefore, motion vectors can be used as a criteria for fusion. This is done by defining a threshold $th1$. We count the number of motion vector with a magnitude greater than th1. If this number is greater than a second threshold $th2$, the motion is considered significant. In this case, the corresponding blocks are set to 1 in the binary mask. The final result is a block-based fusion mask. This algorithm will be referred to as *decoder driven fusion* 2. The thresholds $th1$ and $th2$ are set empirically to 10 and 5, respectively.

It is observed that *decoder driven fusion* 1 performs better than *decoder driven fusion* 2 at low bit rates as shown in Figures 5.30 and 5.31. On the other hand, *decoder driven fusion* 2 performs better at high bit rates. At lower bit rates coding distortions degrade more MCTI side information. Therefore, *decoder driven fusion* 1, which favors the homography-based side information performs better. Conversely, as the bit rate increases, the quality of MCTI side information improves faster. To get optimal performance, both fusion algorithms can be combined. Straightforwardly, *decoder driven fusion* 1 is performed at low bit rates and *decoder driven fusion* 2 at high bit rates. This combination of both fusions is referred to as *decoder driven fusion*.

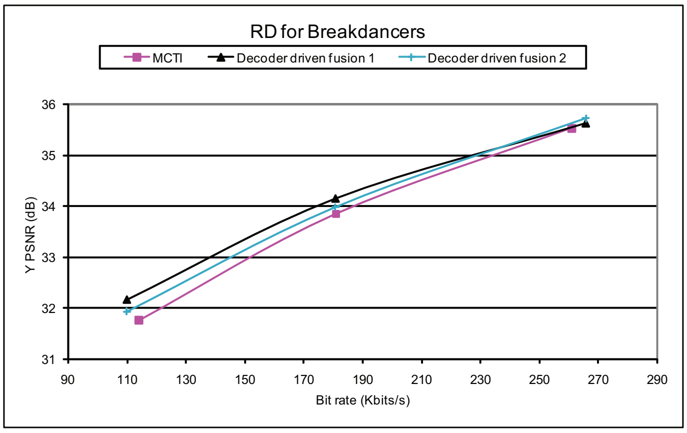


Figure 5.30: RD performance for decoder driven fusions for *Breakdancers*.

Encoder Driven Fusion

The idea is to determine a very good estimate of the WZ frame, which is called the reference, using some help from the encoder. The decision for each pixel (i.e. MCTI or homography) is taken with respect to this reference, as detailed here after.

At the encoder, each pixel of the WZ frame is compared to the ones from the previous and the forward frames. If the one from the previous pixel has a closer value, the binary mask at the pixel position is set to 1. On the other hand, if

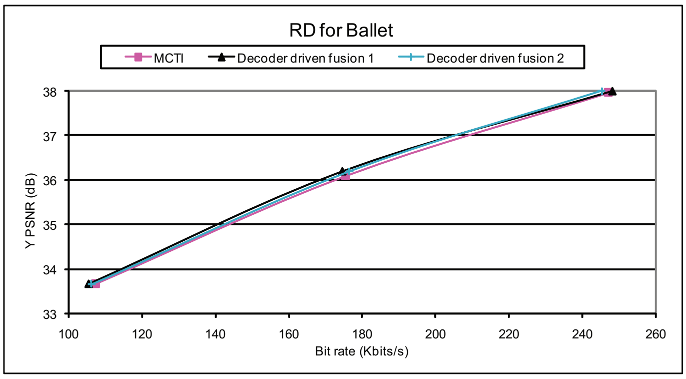


Figure 5.31: RD performance for decoder driven fusions for *Ballet*.

the forward pixel has a closer value, it is set to 0. In other words, it signals the better predictor for the current pixel out of the previous and forward pixels. This process is illustrated in Figure 5.32.

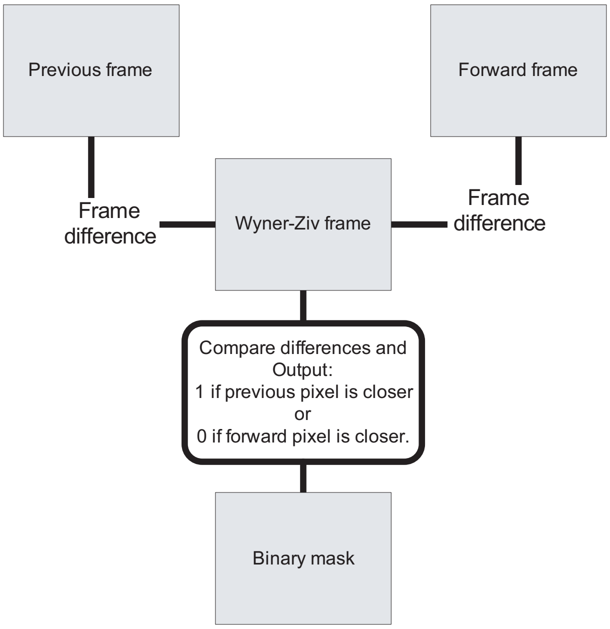

Previous frame
Forward frame
Frame difference
Wyner-Ziv frame
Frame difference
Compare differences and Output:
1 if previous pixel is closer
or
0 if forward pixel is closer.
Binary mask

Figure 5.32: The binary mask generation at the encoder.

The binary mask is encoded using JBIG from the Joint Bi-level Image experts Group [111]. The encoding is preceded by a morphological closing and opening [112]. This eliminates isolated points in the binary mask that would compromise the compression efficiency of JBIG.

At the decoder, the fusion-based side information generation is illustrated in Figure 5.33. The binary mask is recovered and the different side informations, MCTI and homography are computed. The binary mask defines for each pixel which reference to use. 1 in the binary mask means that the pixel values from both side informations should be compared to the pixel from the previously decoded frame. On the other hand, 0 in the binary mask means that the comparison is made with respect to the forward one. This algorithm is referred to as *encoder driven fusion*.

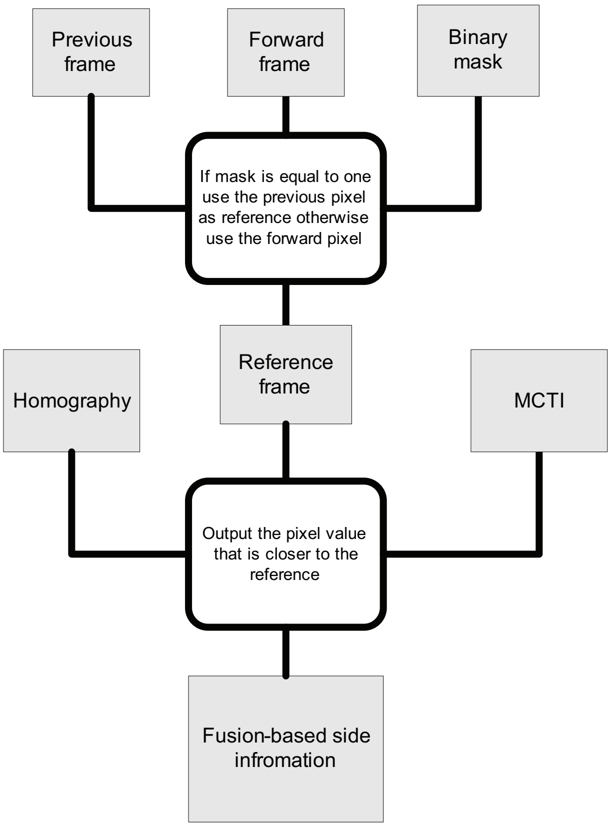

Previous frame
Forward frame
Binary mask
If mask is equal to one use the previous pixel as reference otherwise use the forward pixel
Homography
Reference frame
MCTI
Output the pixel value that is closer to the reference
Fusion-based side infromation

Figure 5.33: The fusion process at the decoder.

5.5.6 Comparison with AVC/H.264

Next, the GT fusion, IMSI and the both fusion techniques, combining MCTI and homography (i.e. the least correlated side informations), are compared to AVC/H.264 Intra, No Motion and Inter Motion. The choice of the Intra and No Motion modes is motivated by the fact they are very close to DVC in terms of encoding complexity. In addition, the DSC theorems state that the performance of a codec that performs joint encoding and decoding (i.e Inter Motion Mode) should also be achievable (asymptotically) by a DVC codec.

For *Breakdancers*, even though the encoder driven fusion is slightly superior to IMSI at low bit rates but overall, IMSI produces the best performance out of the DVC techniques as it outperforms both fusion algorithms superior (Figure 5.34). The performance gap is more significant at high video quality. Nevertheless, IMSI is still inferior to AVC/H.264 in its different modes. This sequence

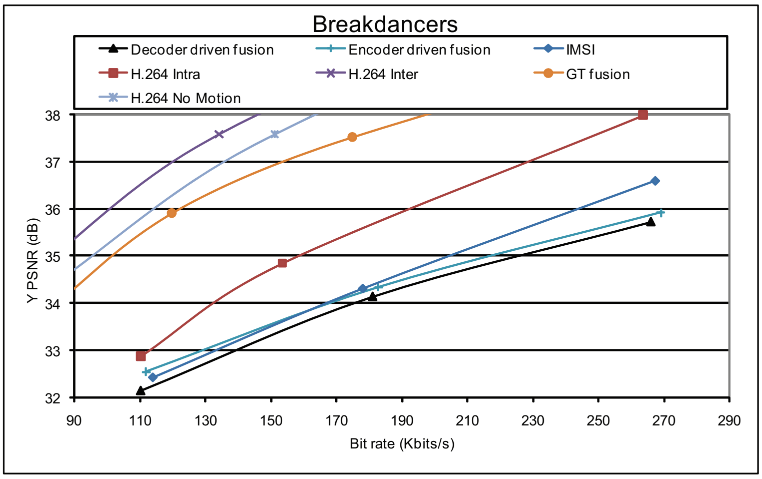


Figure 5.34: RD performance for *Breakdancers* comparing MDVC with AVC/H.264.

is very challenging in terms of motion estimation, which generates a low correlated SI with the WZ frame. This results in a poorer coding performance when compared to conventional codecs.

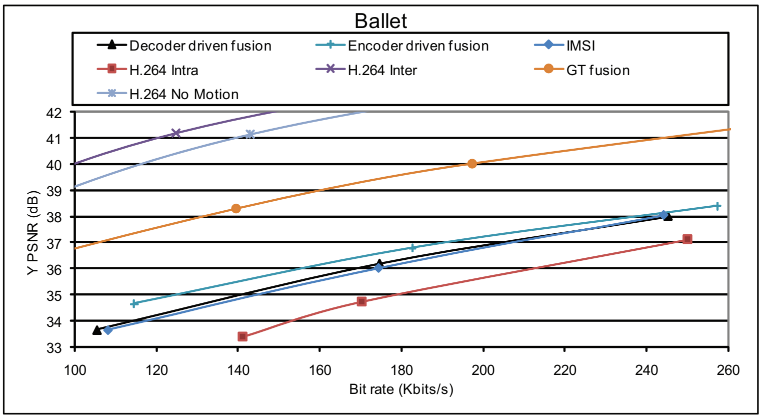


Figure 5.35: RD performance for *Ballet* comparing MDVC with AVC/H.264.

For *Ballet*, IMSI is superior to AVC/H.264 Intra by around 1.0 dB and significantly outperformed by AVC/H.264 No Motion and Inter Motion. Both fusions in this case improve the performance over IMSI. More specifically, the decoder driven fusion's improvement is around 0.25 dB. Moreover, the encoder driven fusion improves the performance even further especially at low and average bit rates by a maximum gap of around 1.0 dB.
For *Uli*, IMSI, which is similar to MCTI in performance, improves the performance over AVC/H.264 Intra by around 3.0 dB. Moreover, it has a poorer performance than AVC/H.264 No Motion and Inter Motion. The fusions do not result in any improvements as the decision is always made in favor of MCTI

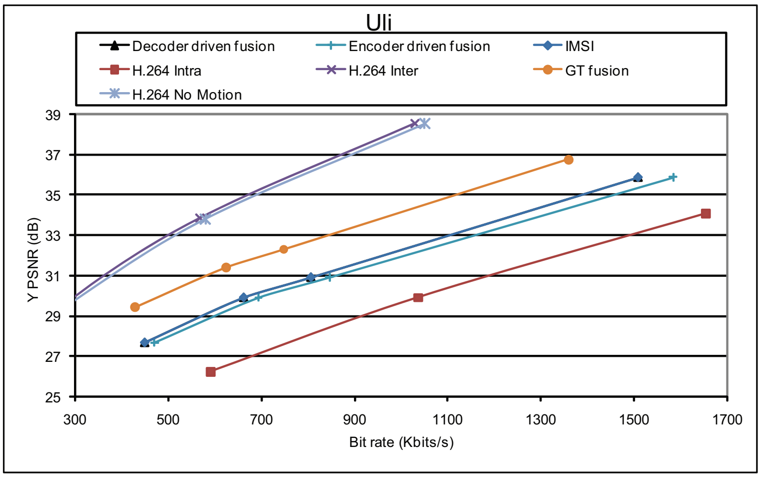


Figure 5.36: RD performance for *Uli* comparing MDVC with AVC/H.264.

for the decoder driven fusion. In other words, performing the fusion in this case is useless for *Uli*. For the encoder driven fusion, the improvement in SI prediction quality is insignificant and since additional rate is spent to send the binary mask, the overall performance drops below MCTI.
Overall, the performance of DVC is superior to AVC/H.264 Intra for two sequences out of three. On the other hand, it has a poorer performance than AVC/H.264 No Motion and Inter Motion for all the sequences, even with the GT fusion. Concerning DVC, IMSI is better for video content with very significant motion occupying a large part of the scene. MCTI is suitable for more or less static video content as it generates highly correlated SI with the WZ frame, resulting in a high compression ratio competitive with conventional coding. For video with average motion, the encoder driven fusion produces the best performance for the DVC compression. Finally, the GT fusion shows that there still a large gap for improvement as it reduces the bit rate for DVC up to 50% for video with significant motion with respect to MCTI.

5.6 Conclusion

DVC is attractive for multiview as it allows for separate encoding of the different cameras in addition to low complexity encoding. Different SI generation techniques are studied for Multiview DVC in this chapter. More specifically, different SI generation techniques are studied for Multiview DVC. For video with significant motion, the proposed IMSI significantly improves the performance over other SI techniques. It is followed by MVME and then MCTI. On the other hand, IMSI is more complex than MVME, which is much more complex than MCTI. For videos with average and low motion, MCTI and MVME improve the RD performance over AVC/H.264 Intra. Nevertheless, MCTI has the advantage of having a similar or better RD performance and being less complex than MVME in this case.
Further, we show that it is possible to reduce up to 50% the bit rate with re-

spect to monoview DVC (i.e. MCTI) with the GT fusion. Nevertheless, the GT fusion requires the original video at the decoder, which is not feasible but it shows the maximum possible gain when the different side informations are ideally combined. It shows as well that MCTI, MVME and DCVP generate highly correlated side informations since they belong to the same block-based category techniques. On the other hand, MCTI and homography represent a good trade-off between performance improvement and complexity increase. Therefore, *fusion* and *encoder driven fusion* combine these two side informations for better compression efficiency than monoview DVC. *encoder driven fusion* results in a better compression efficiency than *fusion* but has the drawback of slightly increasing the complexity of the encoder.

Many improvements are possible over this work. Initially, a better fusion algorithm should be found to exploit the combination of the different side informations without needing the original frame and close the gap on the GT fusion. Moreover, fusion between MCTI and homography should be considered as they produce the least correlated side informations and represent a good trade-off between performance improvement and complexity increase.

Further, the MVME technique is very complex. Therefore, the complexity of this technique can be reduced by using fast motion search techniques such as a multigrid [113] approach instead of a fixed block size in addition to an N-Step [114] search instead of a full search.

Finally, the additional complexity in the IMSI technique can be significantly reduced by selecting the blocks for which the re-estimation is performed as defined in [110]. More specifically, a block is re-estimated in the final SI if the residual error between the initially decoded WZ frame and the initial SI is greater than a ceratin threshold for this block. Otherwise, the block from the initial SI is just copied into the final SI.

Chapter 6

Scalable Distributed Video Coding

6.1 Introduction

Scalable coding is becoming important nowadays in heterogeneous multimedia networks. Different clients on a network might require decoding the same multimedia content at different frame rates, qualities or resolutions, depending on the requirements and the available resources at the client side. For this purpose, scalable coding encodes the content once and enables decoding at different temporal, quality (or Signal to Noise Ratio (SNR)) or spatial resolutions. Scalable coding is attractive for several applications such as surveillance cameras and media browsing.

In this chapter, scalable schemes for image and video coding based on DVC are introduced and compared with scalable conventional coding in terms of compression efficiency. Moreover, these schemes are codec-independent because of the statistical framework of DVC.

Scalable Video Coding (SVC) [115, 116] is introduced by the Moving Picture Experts Group (MPEG) for scalable video compression. SVC is a Motion Compensated Temporal Filtering (MCTF) [117,118] extension of the AVC/H.264 [13] standard using a lifting framework. The temporal scalability in SVC is achieved by the hierarchical B coding structure in AVC/H.264. For spatial scalability, the video is first downloaded at the required spatial resolutions. The corresponding enhancement layers are generated by predictive dependencies with respect to the lower layers. The encoding and decoding processes start at the base layer resolution and progress towards the higher layers. Further, the quality scalability is generated in the same way as the spatial scalability by creating predictive dependencies with respect to lower layers with the same spatial resolution. SVC achieves a very good compression efficiency as it exploits the correlation at the encoder side. In other words, SVC entails high complexity encoding. Furthermore, SVC is based on a deterministic predictive framework (i.e prediction loop), which impairs the performance of the codec in error-prone conditions.

JPEG2000 [12] is a state-of-the-art standard for image coding. Among many features, it enables very efficient scalability. More specifically, spatial scalability is enabled thanks to the use of the Discrete Wavelet Transform (DWT), which results in a dyadic data structure. Quality scalability is ensured by the quality layers, where each one represents a quality increment.

Little research has been performed on scalable schemes based on DVC and it mainly focused on video coding. Tagliasacchi et al. [119] implemented a scalable version of PRISM (Power-efficient, Robust, hIgh compression, Syndrome-based Multimedia coding) [5]. The approach enhances an AVC/H.264 base layer with a PRISM refinement bitstream resulting in a spatio-temporal scalable video codec. It focuses on the case where estimation and most of the motion compensation task is performed at the decoder. Results show that scalable PRISM outperforms non-scalable PRISM and H.263+ Intra, but has a poorer performance when compared to motion compensated H.263+. In fact, since the base layer used AVC/H.264, comparison should have been made with respect to the latter.

A solution to the problem of scalable predictive video coding is introduced in [120] by posing it as a variant of the WZ side information problem. It discusses mainly quality scalability. Results show that the proposed codec is approximately 4.0 dB superior to a naive scalable codec based on a conventional codec. In addition, motion compensation is performed at the encoder which increases its complexity.

Finally, WZ codes are integrated into a standard video codec to achieve efficient and low complexity scalable coding in [121]. Experimental results show improvements in coding efficiency of 3.0∼4.5 dB over MPEG4 FGS [122] for video sequences with high temporal correlation.

Schemes for scalable DVC image and video coding dealing with temporal, spatial and quality scalability are introduced in this chapter. The base layer is generated from the conventional part of the stream, which is used to generate the side information. This is done either temporally by motion compensated interpolation or spatially by a spatial bi-cubic interpolation. The enhancement layer is represented by the WZ bits. Further, the DVC decoding process generates the decoded image or video at full spatial resolution and enhanced quality. The performance of DVC scalable video coding is compared to SVC in error-free conditions. Furthermore, the influence of the base layer quality on the schemes performance is also investigated. For image compression, the comparison is made with scalable JPEG2000. In error-prone conditions, the introduced schemes for image coding are compared to scalable JPEG2000. In this case, the different parts of the JPEG2000 stream are interleaved [123] and then protected with Reed Solomon [16] codes. Moreover, the error resilience tools of JPEG2000 are switched on to enable the recovery from errors present in the stream.

This chapter is outlined as follows. Initially, application of DVC to image and video coding is presented in section 6.2 addressing different scalabilities. Then, the error resilience tools of JPEG2000 are described in section 6.3. The test material and conditions in addition to the simulation results are discussed in section 6.4. Finally, some conclusions are drawn in section 6.5.

6.2 Scalable Distributed Video Coding (SDVC)

In this section, DVC is used to encode images and video in a scalable stream. This approach is shown to be base layer codec-independent [21]. In other words, the base layer can use any conventional codec. In addition, DVC is attractive for

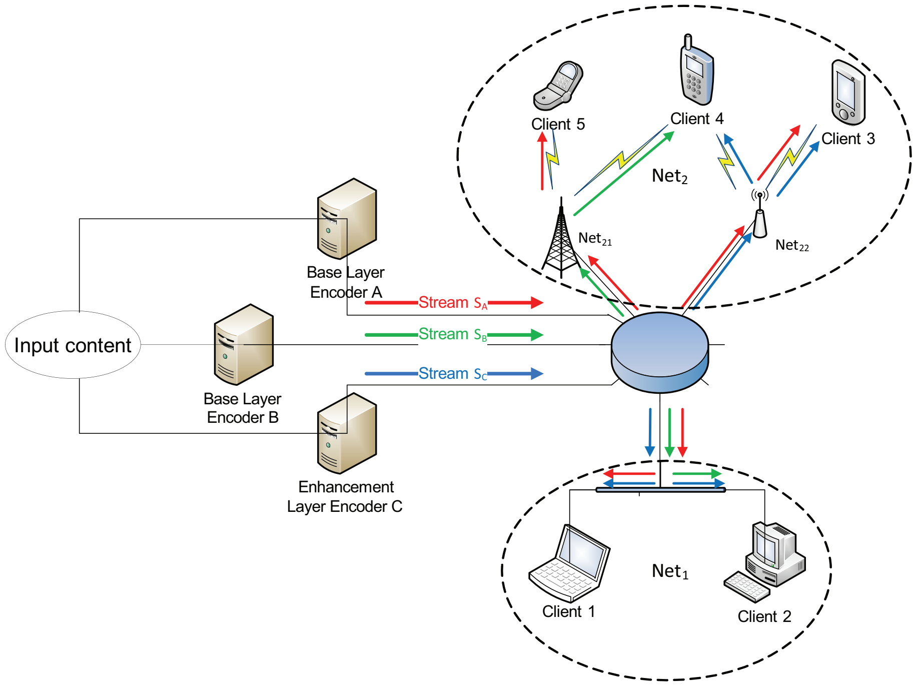


Figure 6.1: Codec-independent scenario.

error resiliency [19, 20]. These advantages are due to its statistical framework. Figure 6.1 shows a scenario where codec-independent scalability can be a very interesting feature. Suppose that we have three content providers A, B and C. A and B will encode the input using two different conventional coders coder$_A$ and coder$_B$ generating streams S$_A$ and S$_B$. Provider C generates the WZ refinement stream S$_C$ for the same input. In network Net$_1$, client 1 and 2 have different conventional decoders dec$_A$ and dec$_B$, respectively. Instead of sending two completely different streams which are both codec A and B compliant, both clients will decode the different base layers and use the same refinement stream. This would reduce traffic in the corresponding network. In network Net$_2$, clients 3, 4 and 5 have the conventional decoders dec$_A$, dec$_B$ and dec$_A$, respectively. Clients 4 and 5 would for example just decode at the base layer quality retrieved from their subnetwork Net$_{21}$. Further, client 3 receives both base and enhancement layers from its subnetwork Net$_{22}$. At the same time, Client 4 has access to subnetwork Net$_{22}$ via which it will receive the enhancement layer at a later stage. Therefore, the codec-independent salability feature offers high flexibility in the way the stream is distributed in the network and helps reducing the traffic at the same time in certain cases.

6.2.1 Scalable DVC for Image Coding

Quality Scalability

Figure 6.2 shows the DVC image coding scheme used for quality scalability. At the encoder side, the input image is simultaneously encoded by WZ and conventional coding. At the decoder side, the conventional part of the stream is decoded generating the base layer. Further, the latter is used as side information along with the parity bits in the WZ decoding process to increase the quality of the base layer.

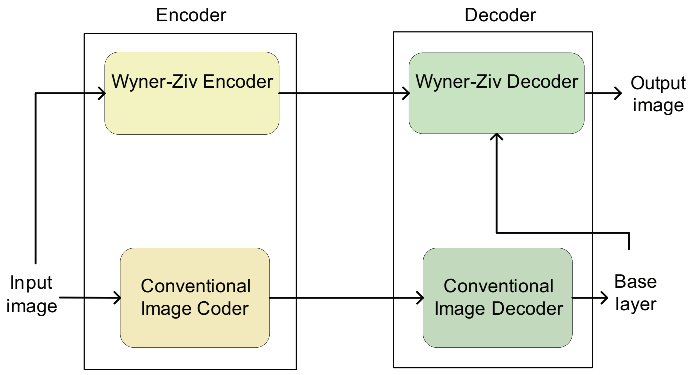


Figure 6.2: DVC architecture for image coding for quality scalability.

There are two ways to control the quality of the decoded image, which are the number of decoded frequency bands and bitplanes. For band level scalability, as more bands are decoded, the quality of the decoded image increases. On the other hand, if all bands are decoded, the quality of the decoded image increases with number of corrected bitplanes. Finally, the best scalability granularity is achieved when a combination of both decoded bands and bitplanes is used.

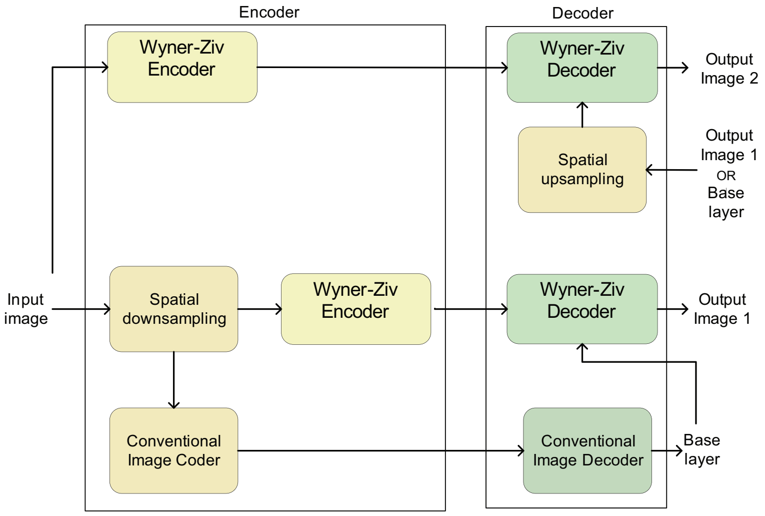


Figure 6.3: DVC architecture for image coding for spatial and quality scalability.

Spatial and Quality Scalability

A scheme for image coding ensuring both spatial and quality scalability is illustrated in Figure 6.3. It is an extension of the quality scalability scheme.

Therefore, a spatial downsampling/upsampling is introduced prior/after the conventional encoding/decoding in the lower branch of the scheme. In parallel, a quality refinement stream is sent for the enhancement of the base layer. Then, the base layer or its enhanced version is used to generate side information to decode the image at the full spatial resolution. Note that the downsampling is preceded by a convolution with a Gaussian kernel to reduce the effect of aliasing, and the upsampling is a bi-cubic interpolation performed in both vertical and horizontal directions.

6.2.2 Scalable DVC for Video Coding

In this section, schemes based on DVC for video coding are described dealing with temporal, spatial and quality scalability. In this case, the DVC schemes for spatial and quality scalability are identical to the ones used for image coding as the same encoding and decoding process is applied to each frame (i.e. image) of the video.

Temporal scalability

The DVC scheme used in this research already ensures temporal scalability. By decoding only key frames, the decoded video has half the original temporal resolution (GOP=2). This makes temporal scalability straight forward. Furthermore, the scheme can be easily extended to n temporal enhancement layers. In this case, a GOP of 2^n is chosen. This results in a base layer with $1/2^n$ the full temporal resolution, whereas the decoding of each enhancement layer doubles the temporal resolution. Figure 6.4 illustrates the decoding process for 2 enhancement layers (GOP=4).

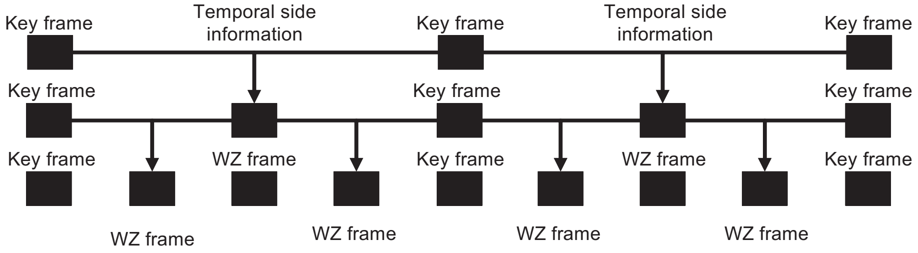


Figure 6.4: Temporal scalability for two enhancement layers (GOP=4).

Temporal, Spatial and Quality scalability

The scheme is depicted in Figure 6.5, by combining the three schemes previously described. First, the odd frames are spatially downsampled and then conventionally encoded in block **A**. When decoded, the latter produce the base layer, $Y_B(2i-1)$ and $Y_B(2i+1)$, which has half the spatio-temporal resolution of the original video. At the same time, WZ bits are sent in parallel to enable the quality enhancement of the base layer to generate $Y_L(2i-1)$ and $Y_L(2i+1)$. Then, temporal side information and DVC decoding are used to generate video at full frame rate and half spatial resolution in block **B**, i.e $Y_L(2i)$. Furthermore, $Y_L(2i-1)$, $Y_L(2i+1)$ (or $Y_B(2i-1)$, $Y_B(2i+1)$) and $Y_L(2i)$ are spatially interpolated and used as side information to generate video at the full spatio-temporal resolution in blocks **C** and **D**. Note that the video can be decoded at half its original temporal resolution and full spatial resolution if blocks **B** and **D** are skipped. This scheme can be easily extended to n layers, since it is a combination of the previous ones.

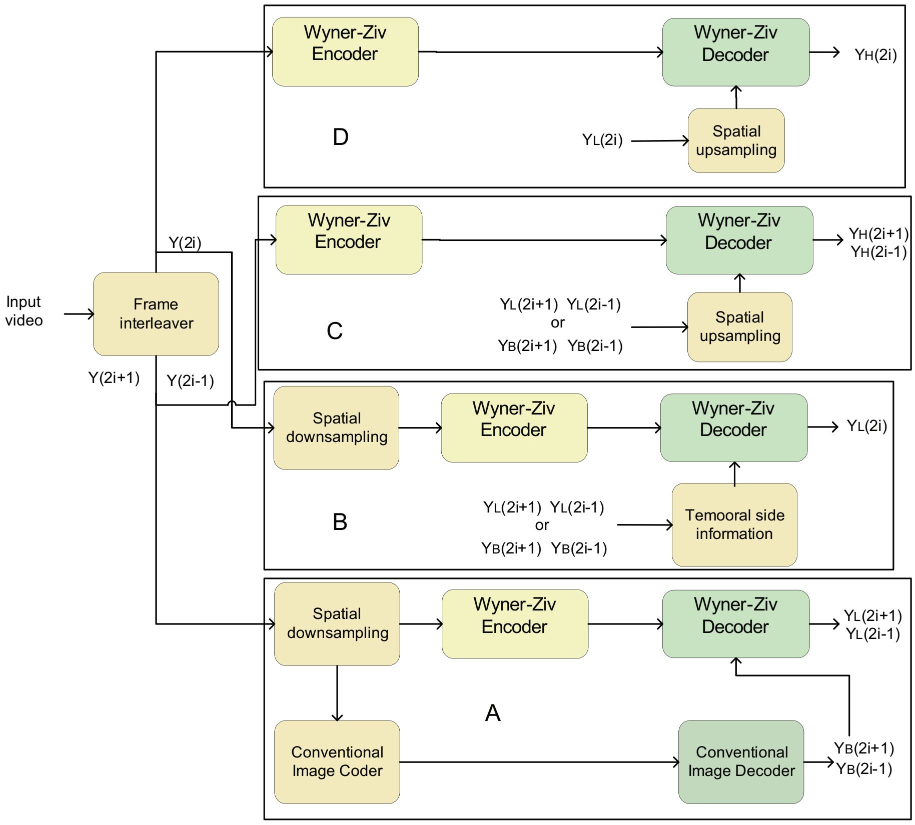


Figure 6.5: DVC scheme for Temporal, Spatial and Quality scalability.

6.3 JPEG2000

The JPEG2000 standard makes use of the Discrete Wavelet Transform (DWT). JPEG2000 supports some important features such as improved compression efficiency, lossless and lossy compression, multi-resolution representation, Region Of Interest (ROI) coding, error resilience and a flexible file format. Figure 6.6 depicts the JPEG2000 fundamental building blocks.

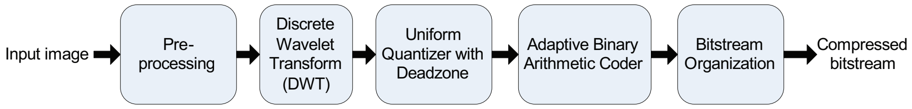

Input image
Pre-processing
Discrete Wavelet Transform (DWT)
Uniform Quantizer with Deadzone
Adaptive Binary Arithmetic Coder
Bitstream Organization
Compressed bitstream

Figure 6.6: The JPEG2000 fundamental building blocks.

For more details on the JPEG2000 standard, already detailed in chapter 3, refer to [12].

To protect the JPEG2000 stream against transmission errors, the bitstream is protected with Reed Solomon [16] codes. Since packet losses are simulated, it is well known that Forward Error Correcting (FEC) combined with interleaving provide more robustness against packet losses. The interleaving prevents from loosing important chunks of the bitstream at the same position but rather bursts the errors over the whole bitstream. The interleaving technique [123] is depicted in Figure 6.7, where each i^{th} word of the interleaved bitstream is constructed from the i^{th} byte of each word from the original bitstream.

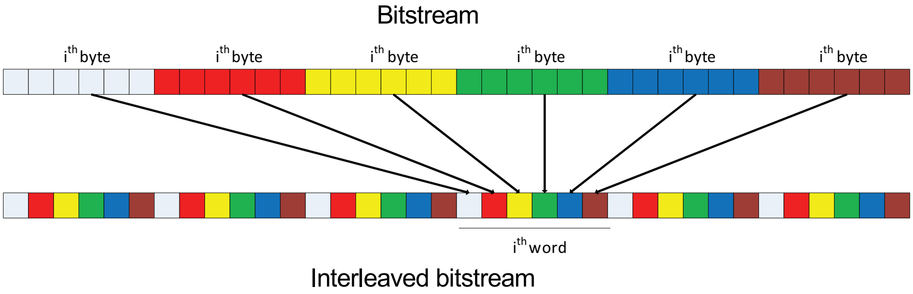


Figure 6.7: JPEG2000 codestream interleaver.

Furthermore, the error resilience tools of JPEG2000 are switched on. A more thorough description of these tools is given in [124] and [125], whereas detailed performance evaluations are presented in [124], [126] and [127]. Note that, although JPEG2000 defines error resilient tools, the procedure that the decoder shall follow in order to cope with the possible presence of errors is not standardized.

JPEG2000 is relying on both resynchronization markers and data partitioning to limit the impact of transmission errors. More specifically, the codestream is composed of packets, with each packet corresponding to a quality layer, a resolution, a component and a precinct. As these packets constitute independently coded units, this data partitioning limits the spread of transmission errors to a great extent. In addition, Start Of Packet (SOP) resynchronization markers can be optionally inserted in front of every packet, as illustrated in Figure 6.8. These markers enable the decoder to resynchronize in the presence of errors. Moreover, the quantized wavelet coefficients are partitioned into code-blocks, each code-block being independently coded using an MQ arithmetic coder. A number of options can be used to strengthen the resilience of the arithmetic

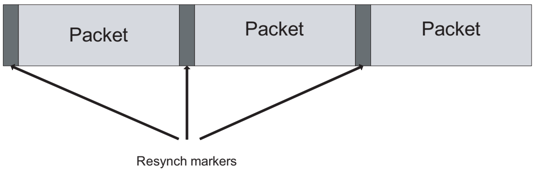


Figure 6.8: SOP resynchronization markers in JPEG2000.

coder. First, the arithmetic coder can be required to use a predictable termination procedure at the end of each coding pass. In addition, the arithmetic coder can be restarted at the beginning of each coding pass. Finally, a segmentation symbol can be encoded at the end of each bit-plane. In this case, if the segmentation symbol is not correctly decoded at the decoder side, an error is flagged in the preceding coding passes. As a direct consequence of these mechanisms, shorter coding passes will entail better error resilience. Therefore, small code-blocks tend to lead to better performance in the presence of errors, unlike the error-free case. While the above tools detect where errors occur, conceal the erroneous data, and resynchronize the decoder, they do not correct transmission errors. Furthermore, these tools do not apply to the image header despite the fact that it is the most important part of the codestream.

6.4 Simulation Results

6.4.1 Test Material and Conditions

The test material consists of the video sequences *Foreman* and *Soccer* with CIF resolution at 15 fps in addition to two monochromatic images *P04* and *P07* with a spatial resolution of 2256x1504. The following codecs are used to run the simulations:

- For the AVC/H.264 Intra coding, the publicly available reference software (JM 11.0) [76] is used with the following settings:
  - High Profile encoding.
  - CABAC for High Profile.
  - The 8x8 transform enabled for High Profile.
  - Disable transform coefficients thresholding.
  - Enable RD optimization.
- For SVC, the publicly available reference software (JSVM 5.1) [128] is used with the following settings:
  - 3 quality layers.
  - 2 spatial layers (QCIF and CIF).
  - 2 temporal layers (7.5 and 15 fps).
- KAKADU version [75] 5.2 is used for JPEG2000 with the following settings:
  - Codeblock size of 64x64.
  - 3 decomposition levels.
  - Visual Frequency Weighting is switched-off. This parameter is used to give good visual appearance. On the other hand, it reduces the RD performance.
  - Target rate control is switched on.
  - Cuse_sop is set true. Use SOP markers in front of each packet at the encoder.

  - Cmodes set to RESTART—ERTERM—SEGMARK. Enable the different error resilience features of JPEG 2000 at the encoder.
  - Use -resilient_sop -resilient at the decoder so that it recovers from and/or conceals errors to the best of its ability.

- The Transform-Domain DVC codec proposed in [36] is used.

In the error-prone case, the feedback channel is switched off and the rates computed in the error-free case are maintained. In other words, we assume the availability of a DVC scheme with an encoder rate control, which is as efficient as the one with a decoder rate control. If the biplane does not converge during the decoding process (i.e. the error probability of the decoded bitplane does not reach a value below 10^{-3}) with the available WZ bits, it cannot use the feedback channel to request additional WZ bits and just uses the bitplane from the SI.
The packet losses are introduced into the base layer and WZ streams with 11 and 51 different packet error patterns [46] for video and image, respectively. The final result is taken as the median of all trials per rate. Moreover, Packet Loss Rates of 5%, 10% and 20% are considered with a packet size of 256 bytes.

Video

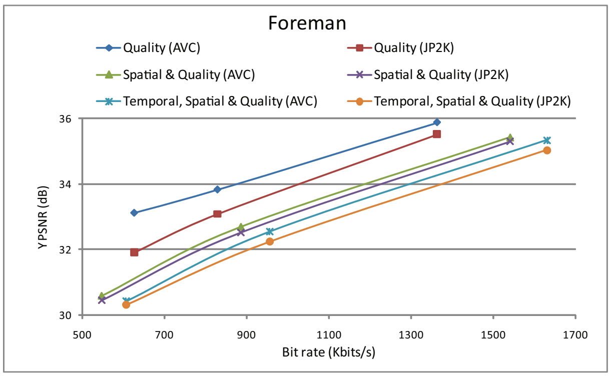
For video, the base layer is encoded using AVC/H.264 Intra at different QPs as lower QP means higher rate and quality. Then, the DVC decoder is run with the corresponding base layer in the error-free case to define the minimum amount of WZ bits for successful decoding. The total rate is computed as the sum of the base layer and WZ rates.
At the encoder, the WZ bits are generated for the quantized DCT coefficients. The quantization is performed using the matrix QI_8

$$QI_8 = \begin{pmatrix} 128 & 64 & 32 & 16 \\ 64 & 32 & 16 & 8 \\ 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 0 \end{pmatrix}$$

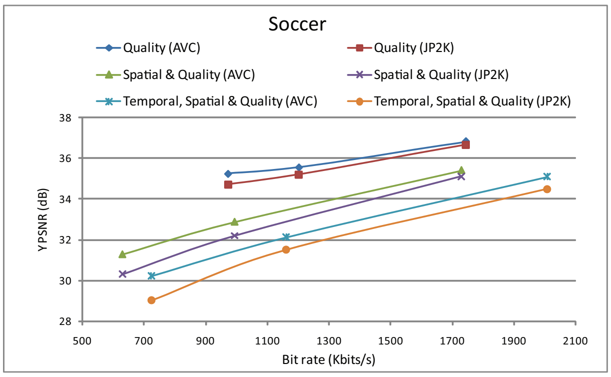
Three RD points are computed per plot. The scalability is achieved by initially decoding the first 4 bands to obtain the first quality layer, which corresponds to the first RD point. Then, the second layer corresponds to decoding 4 additional bands (i.e. a total of 8 bands). This gives the second RD point. Finally, the third layer corresponds to decoding all the bands to obtain the third RD point. The performance of scalable DVC for video coding is compared to SVC with 3 quality layer, 2 spatial layers and 2 temporal layers. Moreover, the influence of the base layer quality on the performance of the DVC schemes in error-free and error-prone conditions is studied. This is only performed for the video schemes as the behavior is similar for images since it is a special case of video, where the number of frames is equal to 1.

Image

The images undergo the same DVC encoding/decoding scenario except that the base layer is encoded using non scalable JPEG2000 and the comparison



(a) *Foreman*.



(b) *Soccer*.

Figure 6.9: The DVC scalable schemes with different base layers and same enhancement layer for *Foreman* and *Soccer*. AVC and JP2K stand respectively for AVC and JPEG2000 encoded base layers.

is made with respect to scalable JPEG2000 in terms of compression efficiency. In error-prone conditions, the JPEG2000 codestream is interleaved and then protected by RS codes. Three protection modes, Weak Protection (WP) (i.e RS(255,179), around 30% overhead of parity bits), Average Protection (AP) (i.e RS(255,153), around 40% overhead of parity bits) and Strong Protection (SP) (i.e RS(255,113), around 56% overhead of parity bits), are considered. The stronger the RS code, the better it recovers from errors with the cost of a larger parity bits overhead.

6.4.2 Error-free conditions

Video

First, simulation results are presented to illustrate the idea of codec-independent scalability. More specifically, the DVC scalable schemes are run with an AVC Intra coded base layer requesting a certain amount of parity bits. This same amount is used to enhance a JPEG2000 coded base layer, which is generated by transcoding the AVC base layer to JPEG2000.

Figure 6.9 depicts the RD performance of the scalable DVC schemes for *Foreman* and *Soccer*. It is is obvious that the performance drops as the number of scalability layers increases. This results in the quality scheme having the best performance and the one ensuring all three scalabilities having the worst performance. Further, it shows that scalability is enabled by DVC for different base layers by the same enhancement layer (i.e. codec-independent scalability).

Hereafter, the base layer quality is varied for the spatial and quality scalability schemes for video as shown in Figure 6.10. Moreover, Tables 6.1 to 6.4 contain the bit rates for the different base layers and the corresponding enhancement layers for *Foreman* and *Soccer*. As the amount of base layer bits is increased (i.e. better SI quality), fewer WZ bits are used and therefore the better the performance. Moreover, the spatial scalability scheme is outperformed by the quality scalability scheme. The reason is that the SI quality for the quality scalability scheme increases rapidly with the base layer rate increase. On the other hand, the SI quality for the spatial scalability scheme saturates rapidly due to the spatial interpolation. It is obvious that scalable DVC lacks of compression efficiency when compared to SVC, which is expected as DVC is known for being inferior to conventional coding for the moment.

Table 6.1: Base and enhancement layers bit rates for *Foreman* for quality scalability.

AVC QP	Base layer Kbps	1^{st} layer Kbps	2^{nd} layer Kbps	3^{rd} layer Kbps
40	255.26	703.29	322.46	232.84
35	471.76	614.35	225.5	159.72
30	843.64	351.5	173.78	115.4
25	1451.62	163.62	146.01	89.32

Table 6.2: Base and enhancement layers bit rates for *Foreman* for quality and spatial scalability.

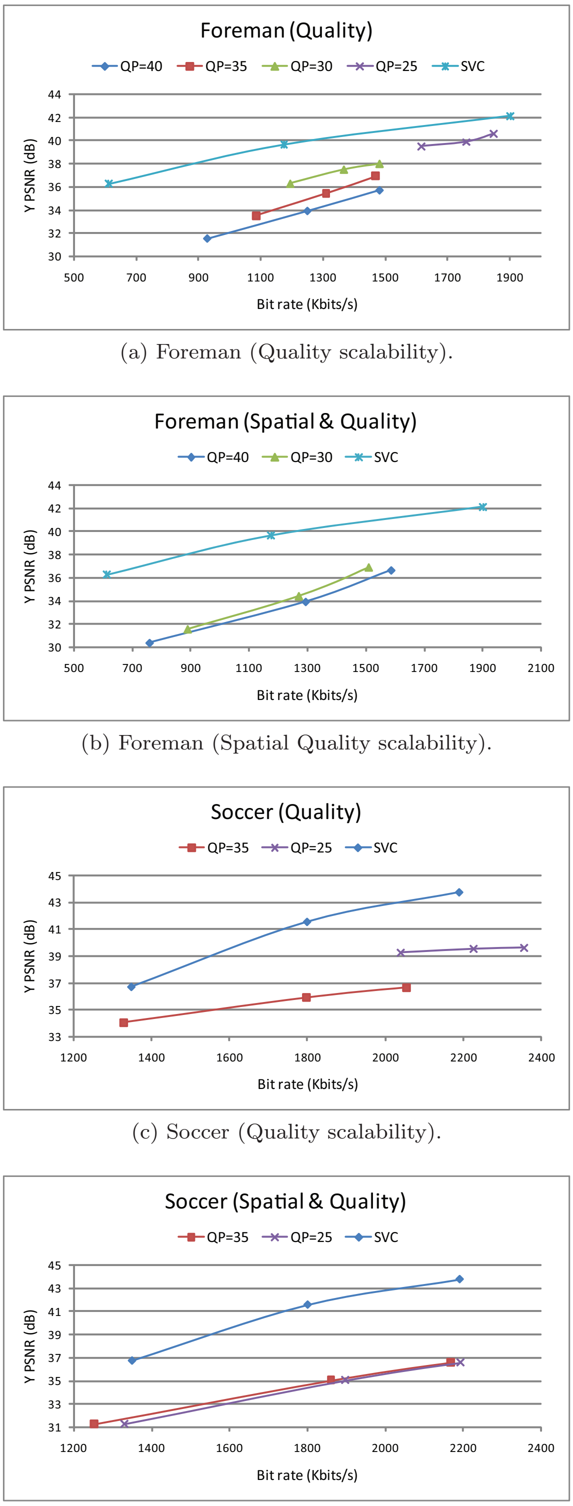
AVC QP	Base layer Kbps	1^{st} layer Kbps	2^{nd} layer Kbps	3^{rd} layer Kbps
40	85.12	675.98	533.53	292.44
30	252.36	639.18	380.84	239.7

Image

The codec-independent scalability feature is shown for images *P04* and *P07* as illustrated in Figure 6.11.

Figure 6.12 shows a comparison between the proposed scalable schemes for image coding and scalable JPEG2000. Moreover, the invested bits in the base and enhancement layers are depicted in Tables 6.5 and 6.6.

With no errors (i.e. 0% PLR), JPEG2000 should be superior to the DVC-based



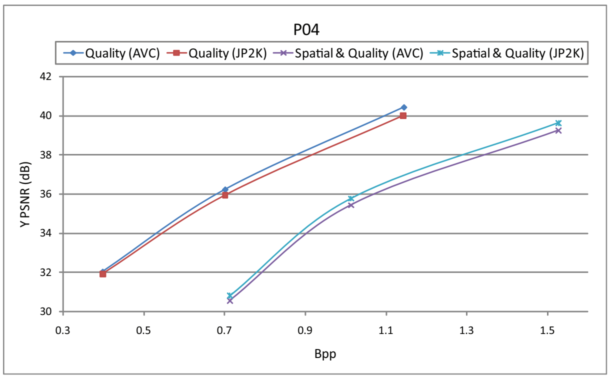
(a) Foreman (Quality scalability).

(b) Foreman (Spatial Quality scalability).

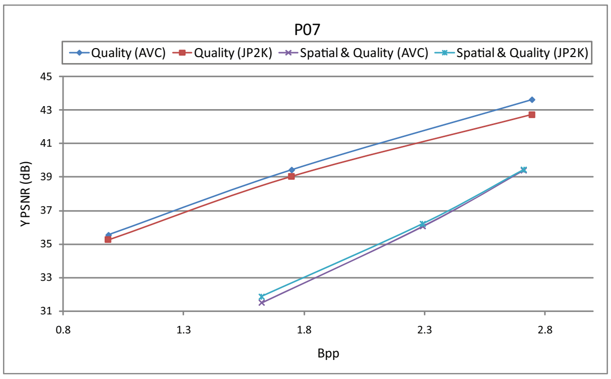
(c) Soccer (Quality scalability).

(d) Soccer (Spatial Quality scalability).

Figure 6.10: The performance of the scalable DVC video coding schemes for different base layer qualities in the error-free case. QP is the Quantization Parameter of AVC/H.264 Intra used to encode the base layer.

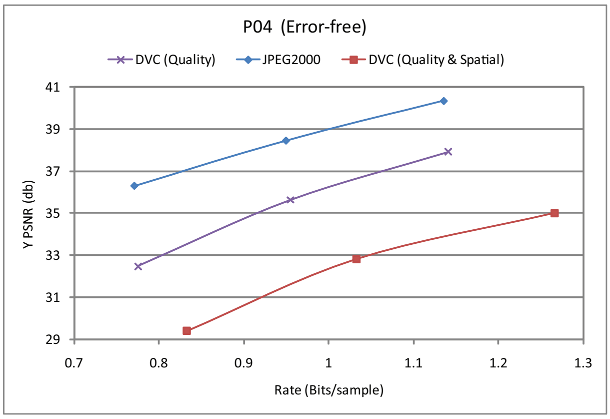


(a) $P04$.

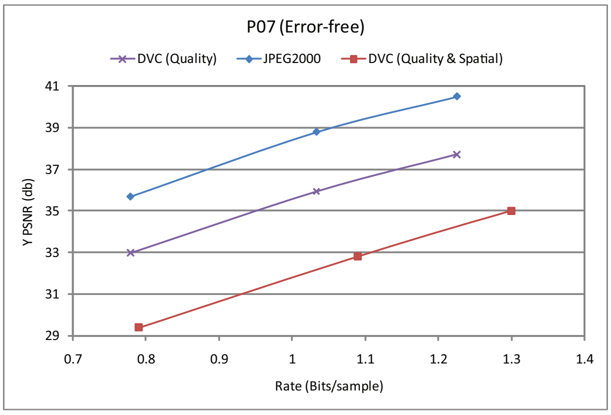


(b) $P07$.

Figure 6.11: The DVC scalable schemes with different base layers and same enhancement layer for $P04$ and $P07$. AVC and JP2K stand respectively for AVC and JPEG2000 encoded base layers.

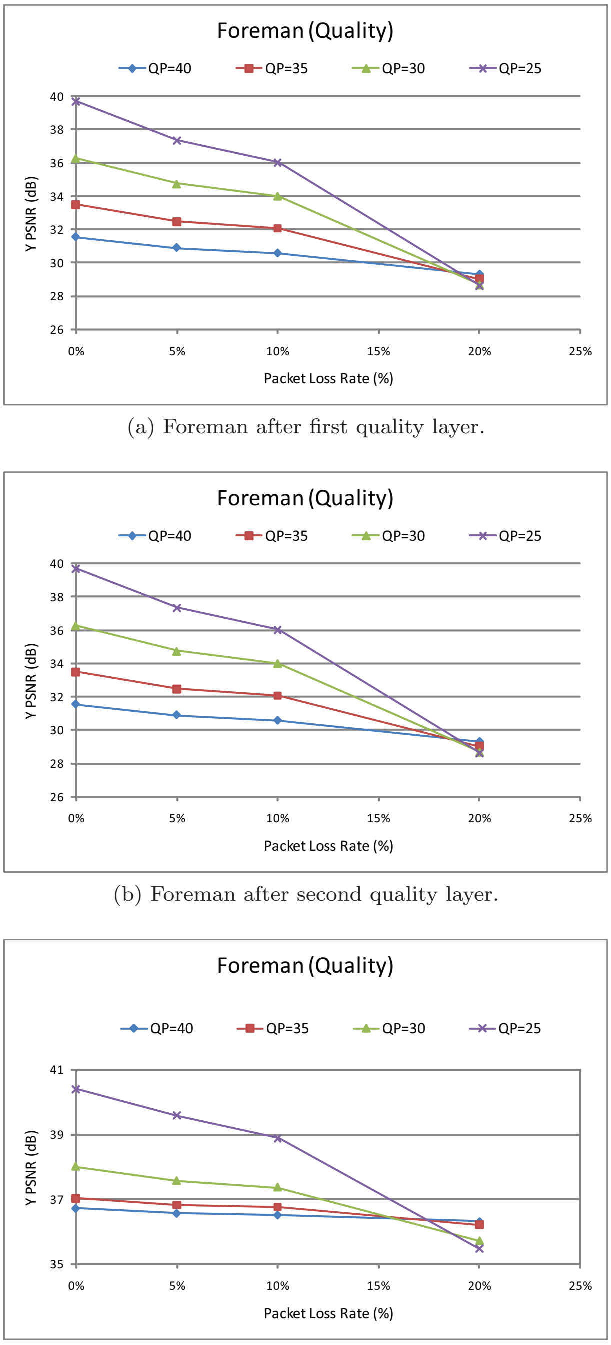


(a) $P04$.



(b) $P07$.

Figure 6.12: Comparison of the DVC scalable schemes with scalable JPEG2000 for $P04$ and $P07$.

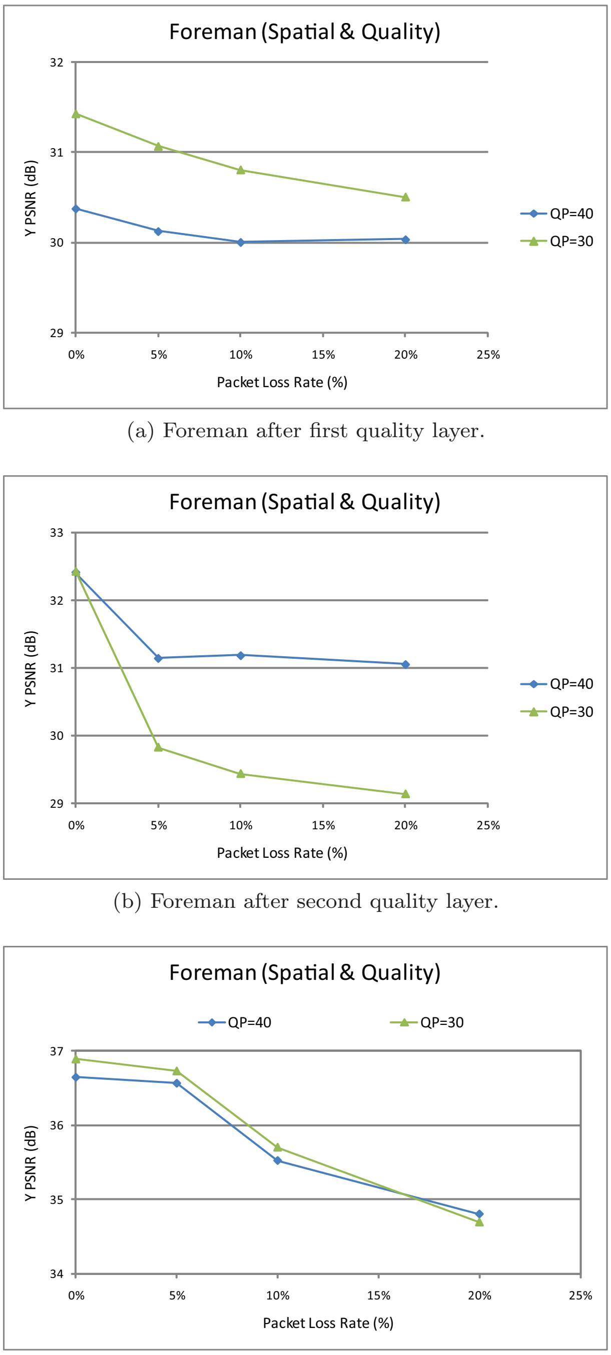


(a) Foreman after first quality layer.

(b) Foreman after second quality layer.

(c) Foreman after third quality layer.

Figure 6.13: The performance of the DVC quality video coding scheme for different base layer qualities. QP is the Quantization Parameter of AVC/H.264 Intra used to encode the base layer.

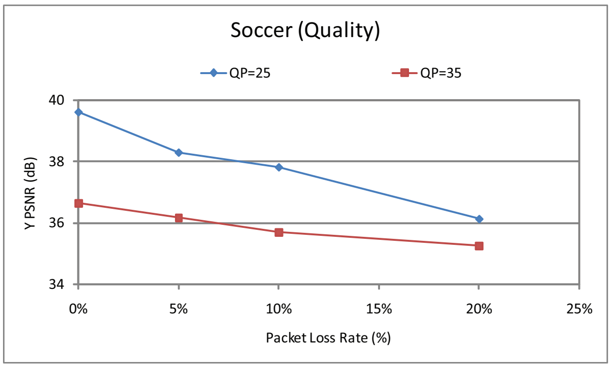


(a) Foreman after first quality layer.

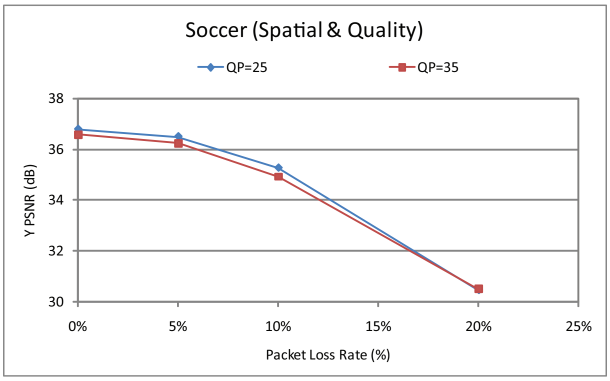
(b) Foreman after second quality layer.

(c) Foreman after third quality layer.

Figure 6.14: The performance of the DVC spatial and quality video coding scheme for different base layer qualities. QP is the Quantization Parameter of AVC/H.264 Intra used to encode the base layer.

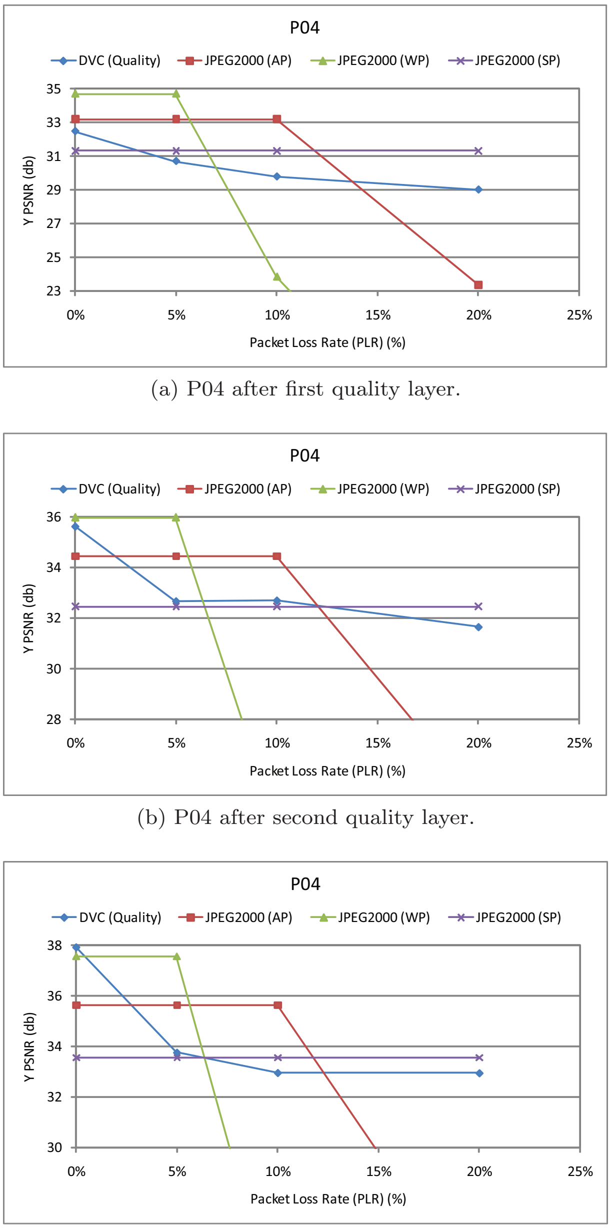


(a) Soccer after third quality layer.



(b) Soccer after third quality layer.

Figure 6.15: The performance of the DVC scalable video coding schemes for different base layer qualities. QP is the Quantization Parameter of AVC/H.264 Intra used to encode the base layer.

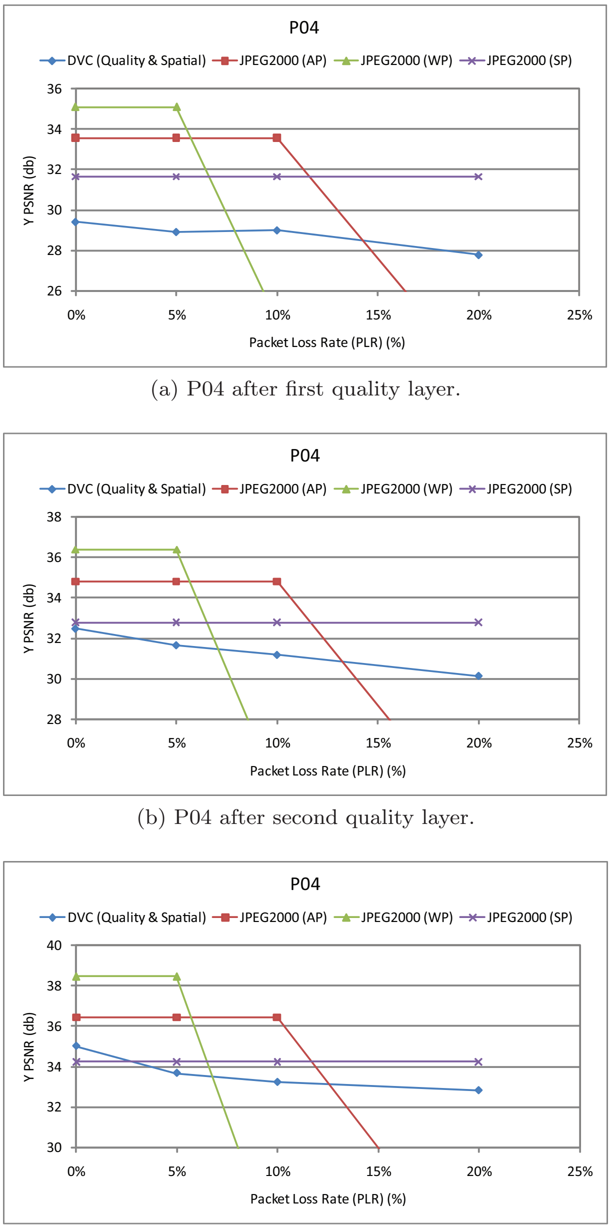


(a) P04 after first quality layer.

(b) P04 after second quality layer.

(c) P04 after third quality layer.

Figure 6.16: Comparison of the DVC quality image coding scheme with RS-protected JPEG2000 for different PLRs at different rates for $P04$. WP, AP and SP stand for Weak, Average and Strong RS protection, respectively.



(a) P04 after first quality layer.

(b) P04 after second quality layer.

(c) P04 after third quality layer.

Figure 6.17: Comparison of the DVC spatial & quality image coding scheme with RS-protected JPEG2000 for different PLRs at different rates for $P04$. WP, AP and SP stand for Weak, Average and Strong RS protection, respectively.

Table 6.3: Base and enhancement layers bit rates for *Soccer* for quality scalability.

AVC QP	Base layer Kbps	1^{st} layer Kbps	2^{nd} layer Kbps	3^{rd} layer Kbps
35	430.43	897.51	470.32	256.91
25	1685.42	354.02	187.3	129.22

Table 6.4: Base and enhancement layers bit rates for *Soccer* for quality and spatial scalability.

AVC QP	Base layer Kbps	1^{st} layer Kbps	2^{nd} layer Kbps	3^{rd} layer Kbps
35	117.83	1135.35	607.3	306.86
25	405.08	924.9	566.81	297.16

schemes if the RS parity bits are omitted as they useless in this case. On the other hand, if the parity bits overhead is large enough, the performance of JPEG2000 can be inferior to scalable DVC as illustrated in Figures 6.16 and 6.17.

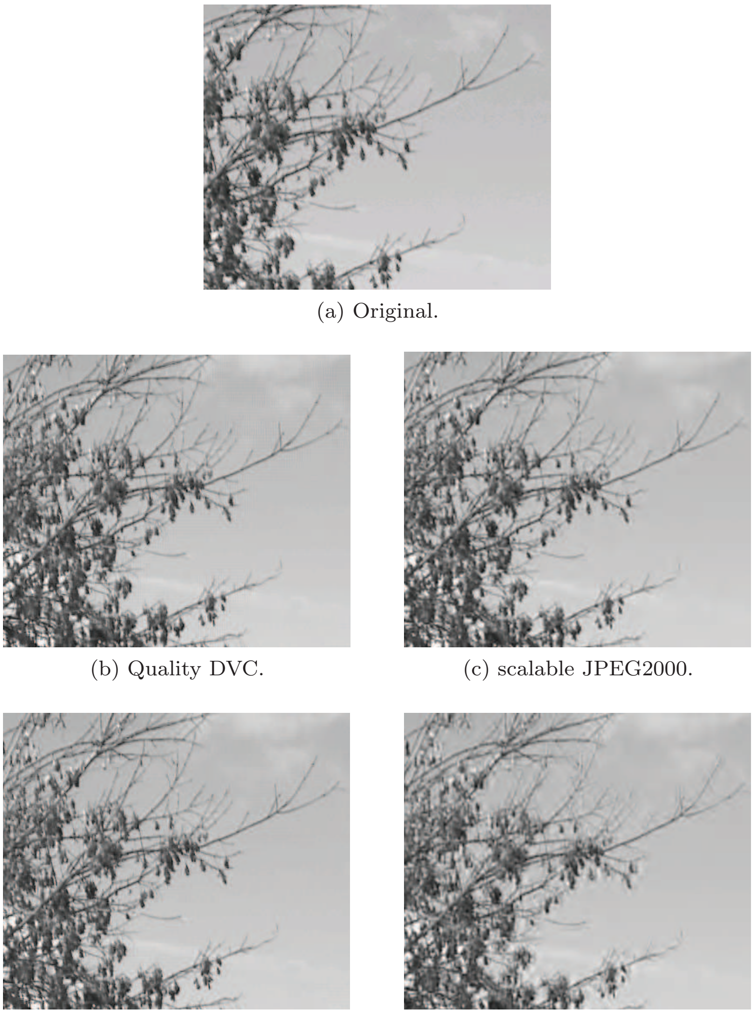
6.4.3 Error-prone conditions

Video

Next, the influence of base layer bit allocation is studied for the proposed scalable DVC schemes for video in error-prone conditions. It is observed that the slope at which the performance decreases with PLRs is greater for higher base layer quality as illustrated in Figures 6.13, 6.14 and 6.15. This is explained by the fact that higher base layer quality corresponds to fewer parity bits. Per consequent, the scheme shows less resistance to the increase in packet loss rate.

Image

The scalability DVC schemes for image coding are studied hereafter in the error-prone conditions by comparing them to scalable JPEG2000 protected with RS codes. The rate for JPEG2000 is computed differently in the error-prone case as the overall rate has to account for the RS parity bits. More specifically, if a Reed Solomon code (n, k) (i.e. $(n-k)$ length parity bits are generated for each k length data) is used and a rate R is targeted, the rate control of JPEG2000 is set to achieve a rate $\frac{k}{n}R$. Thus, the Reed Solomon parity bits rate is $\frac{(n-k)}{n}R$.
It is depicted in Figure 6.16 the comparison between the DVC quality scheme and scalable JPEG2000 with three layers (i.e. three RD points) when packet losses are simulated for $P04$.
First, It is observed that for JPEG2000 WP and AP, the performance tends to



(a) Original.

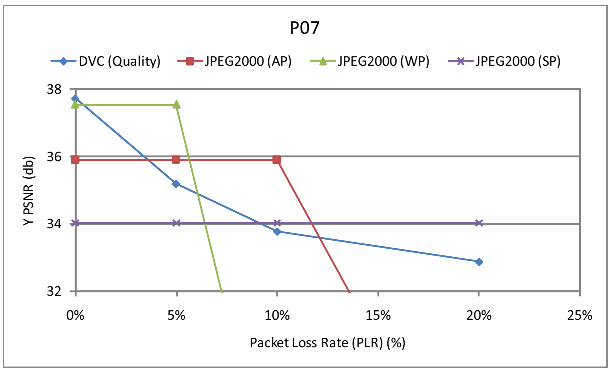
(b) Quality DVC.

(c) scalable JPEG2000.

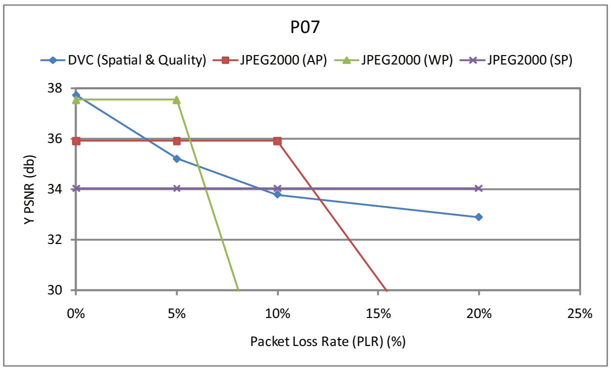
(d) Spatial & Quality DVC.

(e) scalable JPEG2000.

Figure 6.18: Comparison of the visual quality of the decoded image for RS-protected scalable JPEG2000 and the scalable DVC schemes at 10% PLR for *P*04.



(a) P07 after third quality layer.



(b) P07 after third quality layer.

Figure 6.19: Comparison of the DVC scalable coding schemes with RS-protected JPEG2000 for different PLRs at different rates for $P07$. WP, AP and SP stand for Weak, Average and Strong RS protection, respectively.

Table 6.5: Base and enhancement layers bits for $P04$.

	Base layer bpp	1^{st} layer bpp	2^{nd} layer bpp	3^{rd} layer bpp
Quality	0.42	0.35	0.19	0.18
Spatial and Quality	0.15	0.68	0.23	0.2

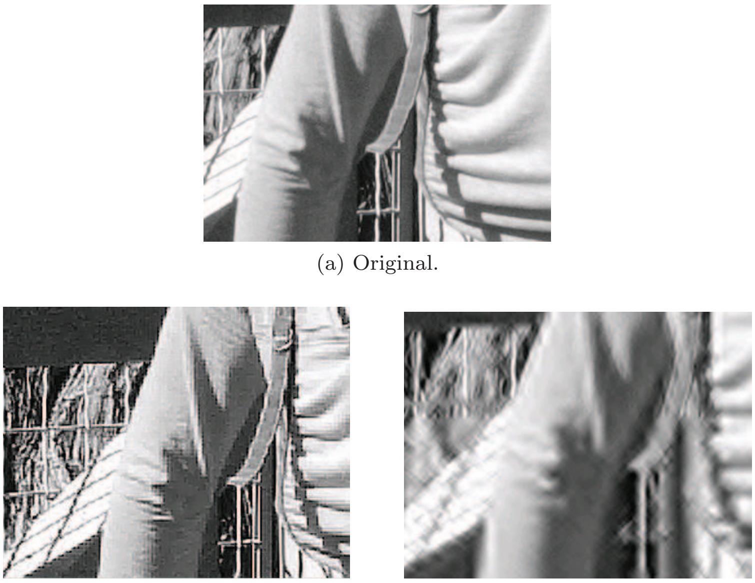
Table 6.6: Base and enhancement layers bits for $P07$.

	Base layer bpp	1^{st} layer bpp	2^{nd} layer bpp	3^{rd} layer bpp
Quality	0.4	0.38	0.26	0.19
Spatial and Quality	0.15	0.64	0.3	0.21

rapidly drop (i.e. cliff effect) after a certain PLR. This is not the case for DVC as it decreases steadily with error rate increase. More specifically, JPEG2000 WP and AP tend to drop rapidly at 10% and 20% PLR respectively as the error correcting capability limit of the RS code is reached. However, the RS code seems to correct all the errors at all PLRs for strong RS protection with the cost of having a huge parity bits overhead. Similar behavior is observed for the spatial and quality scheme as shown in Figure 6.17.

Next, a subjective comparison between the different schemes is performed. First, the visual quality of the decoded image using RS-protected JPEG2000 and the scalable schemes is illustrated in Figure 6.18 for a PLR of 10%, the DVC-based schemes produce a better visual quality image. The spatial-quality scheme performs much better than RS-protected JPEG2000 as the latter produces a highly blurred image. For the quality scheme, the difference is less perceptible. Nevertheless, the DVC-based image has sharper edges and more contrast than the JPEG2000 image.

Figures 6.19 and 6.20 illustrate respectively the objective and subjective quality of the decoded *P07* image at different PLRs comparing the DVC schemes and RS-protected JPEG2000. The image produced by the JPEG2000 decoder is highly blurred while the one by DVC has more contrast but with damaged edges.



(a) Original.

(b) Quality DVC.

(c) scalable JPEG2000.

Figure 6.20: Comparison of the visual quality of the decoded image for RS-protected scalable JPEG2000 and the scalable DVC schemes at 10% PLR for $P07$.

6.5 Conclusion

In this chapter, scalable DVC schemes for image and video coding are introduced. They uses conventional coding to generate the base layer whereas the enhancement layers are represented in the WZ bits. The simulation results show that the proposed schemes are codec-independent. The latter means that base and enhancement layer are completely independent. This offers high flexibility in the way the stream is distributed over the network and helps reducing the traffic at the same time in certain scenarios. In the presence of transmission errors due to packet loss, the DVC schemes show more resistance to error rate increase when compared to conventional coding. More specifically, scalable DVC image coding shows better resistance to the increase in PLR as its performance decreases steadily. On the other, scalable JPEG2000 protected with RS codes for image coding exhibits a cliff effect as its performance drops significantly after a certain limit. Nevertheless, an ideal encoder rate control mechanism is assumed in this work. In other words, the availability of a rate encoder mechanism as efficient as the one with a feedback channel is assumed.

Future work can be pursued into two directions. First, evaluating the same schemes when maintaining the feedback channel for the DVC-based schemes and compare it to the case where it is switched off. Second, incorporate a rate control mechanism at the encoder and perform similar simulations. A poorer performance is expected for the DVC schemes in the error-free case. On the other hand, we would expect a better performance in the error-prone case since the rate will be over estimated when compared to the feedback channel case. In other words, more parity bits are sent towards the decoder.

Chapter 7

DVC Privacy Enabling by Transform Domain Scrambling

7.1 Introduction

Low power video surveillance is identified as one of the most promising applications for DVC in chapter 2. This application requires low power consumption and complexity to keep the cameras simple and small, where using DVC would be very beneficial.

At the same time, the issue of privacy is prominent with the increasing deployment of video surveillance systems due to insecurity concerns because of terrorist attacks and increasing criminality rate. The goal is to conceal video information within regions of interest in video to preserve privacy. The ROI can be defined by a change and unusual event detector or a face detector.

The system in [129] is based on an object-based representation of the scene, where privacy-sensitive objects are masked during display. In [130], an algorithm to protect facial privacy in video surveillance is introduced. The technique preserves many facial characteristics but the face cannot be reliably recognized. Further, privacy filters, expressed using a privacy grammar, are applied to processed video data in [131], preventing access to privacy-sensitive information.

A framework for securing JPEG [11] images is introduced in [132]. It allows efficient integration and use of security tools to ensure confidentiality, integrity verification or conditional access. The latter is performed using a scrambling technique on the DCT coefficients.

In [133], the issue of privacy in video surveillance is addressed for JPEG2000 [12] compression. The privacy is ensured using a transform domain scrambling of regions of interest. It is shown that the technique provides a good security level. Furthermore, the scrambling is flexible and allows adjusting the distortion introduced into the video by varying the DWT decomposition levels to which the scrambling is applied. Moreover, the scrambling precision is defined by the codeblock size. The smaller the codeblock the better the scrambling is fitted to the ROI zone. On the other hand, there is a small loss in the coding performance and a negligible complexity increase.

The problem of scrambling regions of interest for video surveillance to preserve privacy is discussed in [134] and [135]. In addition to JPEG2000, the case of MPEG-4 [122] is also considered. The latter differs from JPEG and JPEG2000 in the different encoding frame modes available in MPEG-4. For JPEG and JPEG2000, only Intra mode is possible. In other words, each frame is encoded on its own without information from its neighboring frames. On the other hand, a frame can be encoded in the predictive mode in MPEG-4. Per consequent, the scrambling has to pay attention not to introduce a drift in the prediction loop. Therefore, it has to be introduced outside of the motion compensation loop. More specifically, AC transform coefficients corresponding to ROI are scrambled by pseudo-randomly inverting their signs, concealing any privacy-sensitive data. Similarly, encryption is used to conceal faces in [136]. A secret encryption key is required in order to invert the process, thus guaranteeing privacy protection.

Further, [137] proposes MPEG-7 cameras, which feature an embedded processor to perform video analysis. The camera does not output an actual video stream, but rather an MPEG-7 compliant descriptor data stream sufficient for video monitoring and surveillance.

Finally, a ROI scrambling technique for AVC/H.264 is introduced in [138]. This is done by either pseudo-randomly inverting or permuting the AC coefficients of the scrambled regions. As for MPEG-4, the scrambling has to be introduced outside the prediction loop by splitting each video frame into two slices, background and foreground slice. The scrambling is only applied to the foreground slice.

In this chapter, a scheme ensuring privacy for DVC is introduced. Moreover, secure AVC/H.264 Intra is used to encode the key frames. For the WZ frames, parity bits are generated for the scrambled DCT coefficients by introducing the DCT coefficient scrambler prior to the WZ encoder. Furthermore, the SI is scrambled as well as it is an estimation of the WZ frame. To decode the video in a clear version, an unscrambling is applied to the finally reconstructed frame. Finally, the comparison of the original and the modified DVC schemes shows that they have a similar RD performance with a negligible increase in rate.

The chapter is structured as follows. Initially, the scrambling is introduced for DVC in section 7.2 addressing privacy for both key and WZ frames. In section 7.3, the security level of the introduced scheme is evaluated. Further, the impact of scrambling on the coding efficiency of DVC is studied by evaluating DVC with and without scrambling in section 7.4. Finally, some concluding remarks are drawn in section 7.5.

7.2 Scrambling for Distributed Video Coding

In this section, the issue of scrambling is initially discussed for key frames using secure AVC/H.264 Intra [138]. Then, the DCT coefficient scrambler is used to preserve privacy for the WZ frames. Finally, different approaches to implement the DCT coefficient scrambler are presented as the end of the section.

7.2.1 AVC/H.264 Intra Scrambling

Since AVC/H.264 [13] Intra utilizes spatial prediction to exploit the redundancy within each frame, this would create coding dependencies between spatially adjacent blocks. Thus, modifying the ROI blocks would induce a drift effect as the scrambling propagates within the frame.
To overcome this problem, Dufaux et al. [138] propose a scrambling that can be effectively applied on the quantized transform coefficients. The latter is done by exploiting the FMO [45] mechanism in H.264/AVC initially developed for error resiliency. More specifically, a slice group map of type 6 is used which specifies an explicit MB assignment of slice group. FMO is used to define two slices composed of MBs corresponding to the foreground and the background, respectively. The fact that each slice is independently encoded ensures that background MBs will not use scrambled foreground MBs for spatial Intra prediction. At the same time, FMO signals to the decoder the shape of the scrambled region required for descrambling. The scrambling is applied prior to entropy coding at the encoder side. At the decoder side, the unscrambling is introduced after entropy decoding. The compressed stream can be decoded by any AVC/H.264 Intra compliant decoder. In other words, unauthorized users are still able to correctly decode the video stream, except for the scrambled coefficients. For more details, refer to [138].

7.2.2 Wyner Ziv Frames Scrambling

To preserve privacy, the DVC scheme is modified as shown in Figure 7.1. At the encoder, the DCT coefficient scrambler is introduced prior to WZ encoding. At the decoder, unscrambling comes down to applying the DCT coefficients unscrambler to the finally reconstructed WZ frame. In addition, since parity bits are generated for the scrambled coefficients at the encoder, the SI DCT coefficients should be scrambled as well. This is because the DCT of the SI is an estimate of the DCT of the scrambled WZ frame. This would prevent spending more rate in the WZ decoding process.

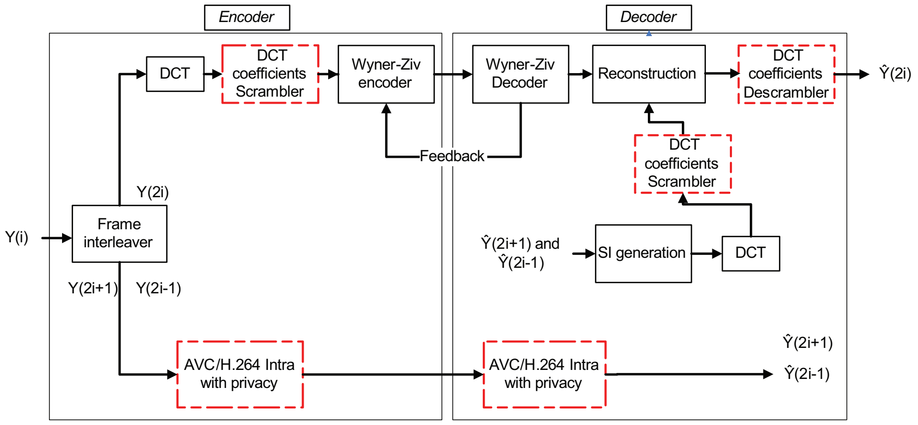


Figure 7.1: DVC scheme with privacy based on transform domain scrambling (GOP=2).

7.2.3 The DCT Coefficient Scrambler

The scrambling and unscrambling are performed via the DCT scrambler and unscrambler, respectively. Hereafter, two approaches of how to implement the scrambling are presented. The first approach consists in pseudo-randomly inverting the signs of the AC coefficients. The second one consists in pseudo-randomly permuting these coefficients. Furthermore, both scrambling approaches are driven by a Pseudo Random Number Generator (PRNG) initialized by a seed $\boldsymbol{s}$, which is encrypted and constitutes the secret key. The latter is required as the decoder to perform the descrambling.

Random Sign Inversion

In this case, the scrambling process consists in pseudo-randomly flipping the sign of the AC coefficients within each 4x4 scrambled block. This should not have a negative impact on the coding efficiency as the magnitude of the coefficients is maintained but only the signs are changed.
The AC coefficients $DCT[i]$ with $i = 1..15$ are scrambled by generating a pseudo-random binary sequence $b_i \in \{0, 1\}$ such that

$$DCT[i] = \begin{cases} -DCT[i] & \text{if } b_i = 1 \\ DCT[i] & \text{if } b_i = 0 \end{cases}$$

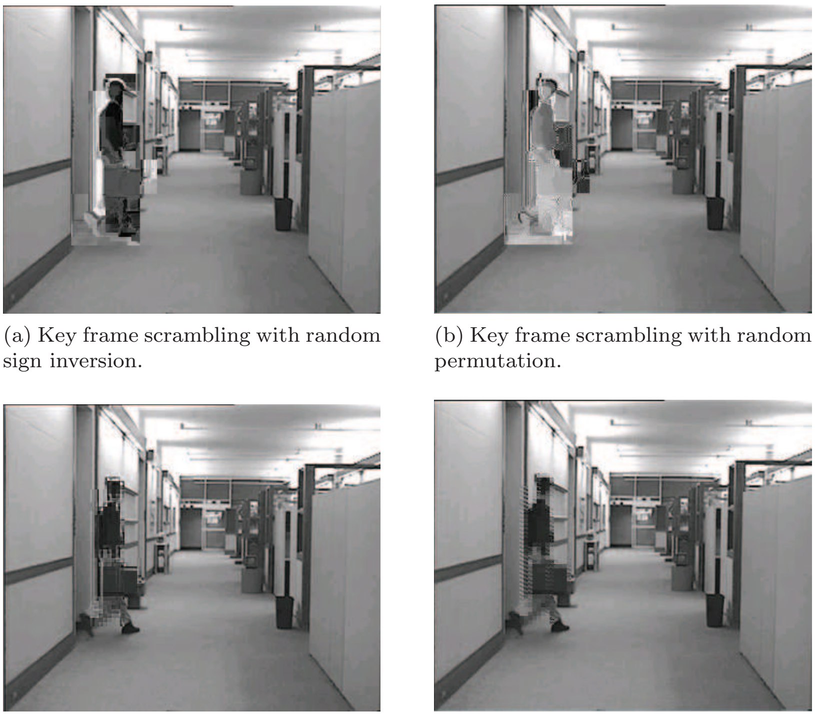
. The increase in complexity with this approach is negligible as it consists in a simple sign inversion.
The unscrambler is equivalent to the scrambler in this case as sign inverting of the scrambled coefficients restores the original signs of the coefficients as long as the same pseudo-random binary sequence is generated at the decoder, which requires the same seed $\boldsymbol{s}$.

Random Permutation

An alternative approach to perform the scrambling is by applying a pseudo-random permutation to rearrange the order of AC coefficients of the 4x4 scrambled block. The Knuth shuffle [139] is used to generate a permutation of the coefficients. More specifically, a scan of the AC coefficients through each position i from 1 to 14 is performed and the element at position i is swapped with a pseudo-randomly chosen element from positions $i + 1$ through 15. A greater loss in compression efficiency is expected with this approach when compared to the previous one as both sign and magnitude of the coefficients are changed. Moreover, the increase in complexity is greater as well. The unscrambler in this case corresponds to the inverse permutation.

7.3 Security Issues

The scrambling is applied on a 16x16 block-basis. When the scheme is attacked by brute force, the latter will require a maximum of $2^{(16.16)-1} = 2^{255}$ iterations per 16x16 block in case of random sign inversion. When the scrambling is applied by random permutation, the brute force attack would require 255! iterations, which is much greater than 2^{255}. This makes the scrambling more secure with the random permutation.



(a) Key frame scrambling with random sign inversion.

(b) Key frame scrambling with random permutation.

(c) Wyner Ziv scrambling with random sign inversion.

(d) Wyner Ziv scrambling with random permutation.

Figure 7.2: The scrambling perception for both key and WZ frames.

The scrambling is applied only to AC coefficients to have the visual perception of the scrambling as shown in Figure 7.2, where the privacy is ensured and at the same time the scene is understood. The amount of distortion introduced is controlled by the total number of AC coefficients to which the scrambling is applied to. The smaller this number, the less the introduced distortion is.

7.4 RD Performance

In this section, the effect of the scrambling on the compression efficiency is studied. For this purpose, the sequence *hallmonitor*, CIF@30 fps, is used to evaluate the RD performance of both DVC schemes, with and without privacy. More specifically, the following two scenarios are compared. The first one is where no scrambling is applied, which corresponds to the original scheme. The second one is where scrambling and unscrambling are applied respectively at the encoder and decoder sides.
It is obvious from the plot in Figure 7.3 that the scrambling slightly decreases the RD performance with a greater loss in the permutation case. This is because only the sign of the coefficients are changed in the sign inversion case. On the other hand, both sign and magnitude of the coefficients are changed in the permutation case.
In the random inversion case, the increase in rate is around 2.6% and 1.0% at

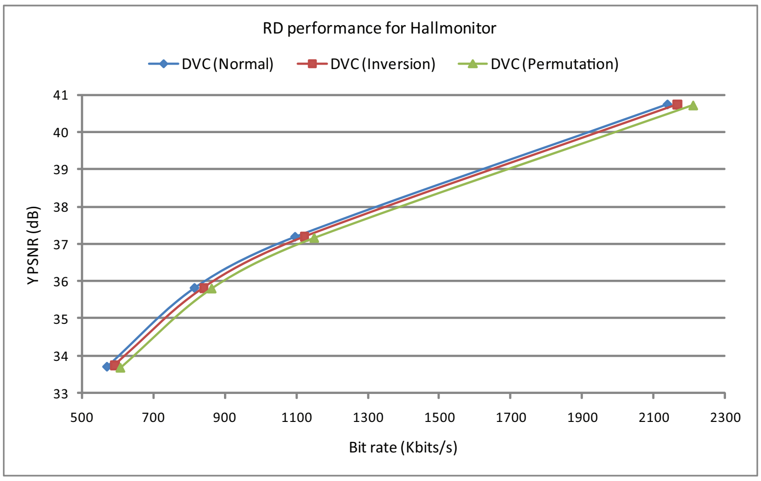


Figure 7.3: The RD performance of DVC with and without scrambling.

low and high bit rates respectively. It is around 5.3% and 3% in the random permutation case.

7.5 Conclusion

In this chapter, an efficient scrambling scheme for DVC is introduced, based on transform domain scrambling. For the key frames, secure AVC/H.264 Intra is used. Due to the dependencies between adjacent blocks in Intra coding, each key frame is split into two slices, background and forward slice. The scrambling is only applied to foreground slice to avoid drift in the background slice. For the WZ frames, the parity bits are generated for the scrambled version of the input frame. At the decoder, unscrambling is applied to the finally reconstructed frame. Moreover, scrambling of the SI is also necessary as it represents an estimation of the WZ frame. The scrambling can be applied as a simple random sign inversion operation or a random permutation. The scrambling preserves privacy with a sufficient level of security and a flexible scrambling level. It is shown that the scrambling has a negligible impact the compression efficiency of DVC.

Chapter 8

DVC Demonstrator

8.1 Introduction

This chapter describes the conception and the implementation of a DVC demonstrator based on the DISCOVER software [17]. This demonstrator is defined in a setup being as close as possible to a complete real application scenario where software complexity, hardware requirements, network transmission and practical deployment aspects are taken into account. In this scenario, two main entities are highlighted, the client demonstrator and the server demonstrator running the DVC encoder and decoder respectively. A careful study of the potential limiting factors of these entities is performed, namely the real time encoding, the presence of a feedback channel and the real time decoding. The latter is the target of an exhaustive algorithm optimization effort realized on the DVC decoder software. A complete description of the graphical interfaces created is provided, to justify the inclusion of some software and to provide some experimental results of the demonstration.

This chapter is organized into five different sections. Section 8.2 provides a list of requirements for the demonstration scenario. A complete description of the infrastructure needed for the demonstration purpose is presented. More precisely, a transmission scenario between the client and the server demonstrator is described. The choice of the transmission protocol used in the transmission of the DVC compressed bit stream is justified. Moreover, a detailed explanation of the composition of both demonstrator client and server is exposed. In Section 8.3, the work performed on the optimization of the DVC decoder software is described. In this section, a study to find the main bottlenecks that prevent real time decoding is presented, followed by a list of proposed solutions. The results achieved provide significant gains in terms of decoding speed with respect to the initial software. However these results show that decoding is still far from being performed in real time. Section 8.4 describes the practical implementation of the demonstrator's client and server. Moreover, the use of some libraries is justified with more attention to the x264 [140] encoding library used in the client demonstrator. In addition, a complete description of both client and server Graphical User Interfaces (GUI) is outlined in Section 8.4. Finally, the main conclusions about the implemented demonstration scenario are presented in section 8.5.

8.2 Demonstrator Design

The goal of this section is to provide a detailed description of the demonstration scenario. It should exhibit a complete real application scenario in which WZ coding can be used. In particular, the demonstrator should capture a video sequence with multiple video cameras, code the view, simulate the network transmission and finally, decode and display the decoded video stream using a dedicated server.
In Figure 8.1, a multiple camera architecture is described. This architecture is closely related to the applications identified as most promising, notably wireless low-power surveillance, wireless video cameras, and visual sensor networks. In terms of functionality, it is possible to separate the demonstrator's architecture into the following entities:

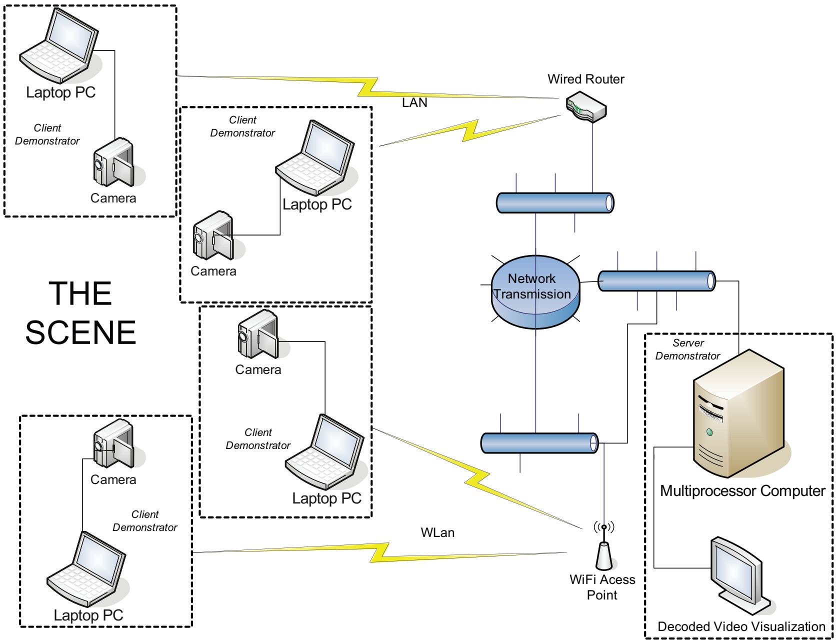


Figure 8.1: Demonstration scenario.

- **Camera** - It is a digital video source connected to a laptop PC by a Universal Serial Bus (USB) cable.
- **Client PC** - It is connected to the camera and to the local network and runs the client demonstrator.
- **Multiprocessor Computer** - It is connected to the local network and runs the server demonstrator.
- **Network Transmission** - For the transmission of the compressed streams, a shared resource implementing the NetBIOS/TCP [141] protocol in the Laptop PC is used.

The following two subsections present the client and the server demonstrator.

8.2.1 Client Demonstrator

The client demonstrator integrates into one entity the camera and the laptop PC running a graphical application. Furthermore, the client PC is connected to the local network, as shown in Figure 8.1. The functional diagram of the client demonstrator is depicted in Figure 8.2.

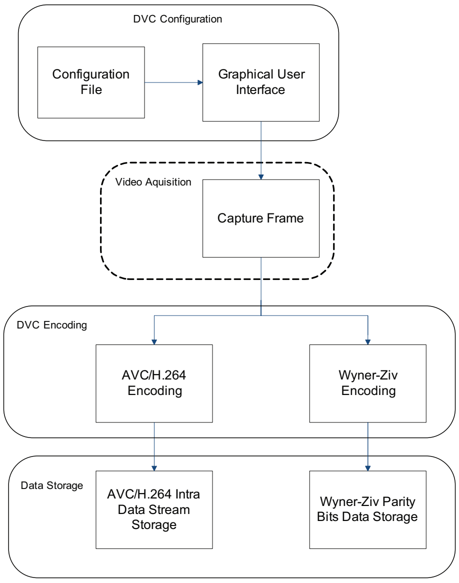


Figure 8.2: Functional diagram of client demonstrator.

The client demonstrator controls all the aspects of the encoding demonstration scenario:

- **Video Acquisition** - Controls the acquisition of video through a connected capturing device.
- **DVC Configuration** - The DVC encoding parameters are set trough a configuration file and directly in the graphical interface.
- **DVC Encoding** - This application includes the DVC encoding software.
- **Data Storage** - The data generated by the DVC encoding software is stored locally in the client demonstrator, on the hard disk of the Client PC and shared with the multiprocessor server computer via the network.

8.2.2 Server Demonstrator

The server demonstrator integrates into one entity the server computer running a graphical application. This server is a multiprocessor machine. Furthermore, the server is connected to the network as shown in Figure 8.1. The functional diagram of the server demonstrator is depicted in Figure 8.3.
This application controls the following aspects of the decoding demonstration scenario:

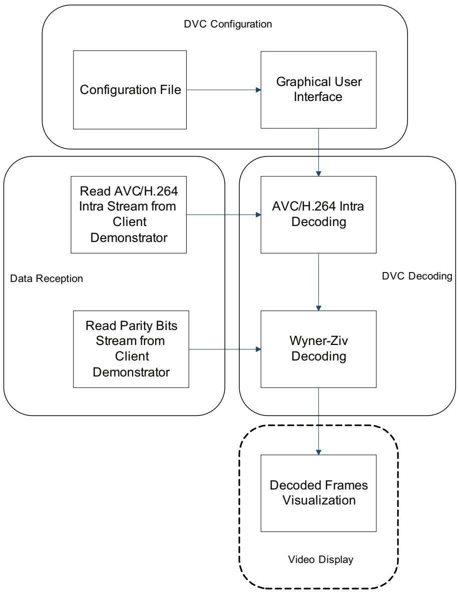


Figure 8.3: Functional diagram of server demonstrator.

- **DVC Configuration -** The DVC decoding parameters are fixed using a configuration file and directly in the graphical interface.
- **Data Reading -** In the DVC decoding process, the server demonstrator reads the data from the shared hard disk on the Client PC, simulating the network transmission upon request with the feedback channel.
- **DVC Decoding -** This application includes the DVC decoding software.
- **Video Display -** In this application, it is possible to have access to the decoded video trough a pop-up window, either during the execution of the application or later for the user's inspection.

8.2.3 Network Transmission

For the transmission of the AVC/H.264 Intra stream and the WZ parity bits, the available possibilities are TCP/IP, UDP and a shared resource on the network (NetBios/TCP). The main advantages and disadvantages of these protocols are:

- **TCP/IP -** The main advantage of this protocol is the guarantee of packets delivery but without exploiting the available bandwidth in the network.
- **UDP -** The main advantage of this protocol is the fact that it performs a best effort to deliver data, but without guaranty of packets delivery.
- **NetBios/TCP -** This protocol provides a simple and reliable way to establish a connection between computers in a local network but without having much control since it is an operating system specific protocol.

The NetBios/TCP approach is selected for the demonstration since realtime decoding is not accomplished by the DVC codec and video streaming is out of the scope of this research.

The adopted architecture to transmit data between the client and the server demonstrator is depicted in Figure 8.4.

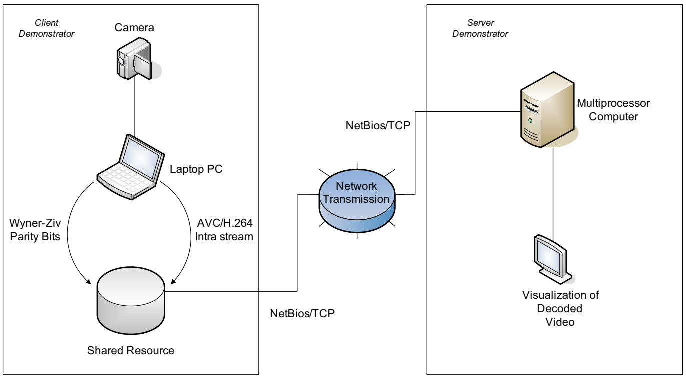


Figure 8.4: Network transmission scenario.

The client demonstrator produces two streams, the key frames encoded in a traditional way, e.g. using the AVC/H.264 standard and the parity bits generated using the WZ approach. Both streams are stored locally in the hard disk of the Client PC which is being shared on the network. On the other side, the server demonstrator reads the key frames and the parity bits directly from this shared resource, thus simulating a feedback channel.

8.3 Video Decoding Optimization

One of the major limitations of the DVC software is the decoder's high complexity. The key frame and WZ decoding are the two major components of the decoder. The biggest bottleneck is the WZ decoding part since it is based on an expensive iterative implementation which has a negative impact on the software in terms of decoding time. This section discusses the algorithmic complexity of the DVC decoding software.

8.3.1 Decoder Profiling

In order to identify the major limitations of the software, a profiling of the DVC decoder is performed. Figure 8.3.1 shows the results obtained.

The profiling results show that the major part of the algorithm calls is in the WZ Decoding (63% of total calls) and SI Generation (36%). The other remaining functions of the algorithm represent 1.0% of the calls. The goal is to reformulate the implementation of these two critical parts of the software to achieve faster decoding.

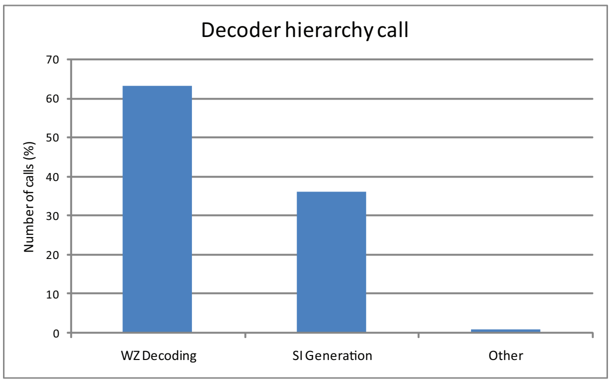


Figure 8.5: Profiling of the decoder software.

8.3.2 Optimization Solutions

Different means are considered in order to reduce the decoding time, including the usage of inline functions, multithreading, loop unrolling, architecture based compiler and look up tables. The properties of these different methods are first briefly recalled before discussing their implementation within the DVC decoder.

Multithreading / Parallelization

By decomposing the decoding algorithm and distributing it over different threads, substantial speed gains can be achieved. Referring to the algorithmic video decoding process, it is possible to apply multithreading at different levels, namely at the frame level (i.e. by decoding several WZ frames concurrently)and at band level (i.e. by decoding several frequency bands at the same time).

Loop Unrolling

Loop unrolling consists in reformulating one software loop - or possibly several interspersed loops - such that the software code gets executed linearly without tests / jumps. The benefit of doing so is that the computational context gets linearized, so that the underlying parallelism can be used without suffering from processing interruptions caused by the tests and jumps related to the loop control. Furthermore, the fine grain control of the instructions executed gets more optimal since there is no need to execute dummy operations for internal pipe synchronization purposes when reaching the jump instruction as occurring otherwise. Globally speaking, loop unrolling helps in making the code execution more regular. Obviously, the overhead caused by the loop control is also avoided at the price of a substantial increase of the program code size.

Architecture-based Compiler

There exist compilers that enable the production of object code taking into account the detailed architecture of the target processor, so as to optimize the usage of resources and execution time. Some compilers have special option flags to steer the automatic optimization.

Inline Functions

Using inline functions in the code of the algorithm can lead to some gains in terms of speed, especially if these functions are often executed. Inline functions are expanded at the calling block. In this case, the function is not treated as a separate unit like other normal functions. At the end, it comes down to the compiler to decide if a function is qualified to be an inline function, depending on the amount of code it includes. The keyword *inline* is used to declare a function as inline in C++.

Look-up Tables

Using look-up tables to replace runtime computations with simple lookup operations in data structures (arrays, etc.) can provide significant gains in terms of algorithmic speed since retrieving a value from memory is faster than undergoing an expensive computation. Usage of look-up tables gets advantageous when repeated retrieval of certain values is needed, and/or when the resort to look-up tables enables a substantial improvement of the software regularity.

8.3.3 Implementation and Evaluation

The various approaches discussed above have been partly combined and evaluated for different configurations that are being compared hereafter. Out of the methods listed above, multithreading / parallelization offers the best performance on a multiprocessor computer architecture.

Multithreading on Frame Level, Compiler optimization, Look-up tables

The DVC decoder software is modified with the objective of putting the decoding bulk of each WZ frame in a different thread to make use of all available processors belonging to the server.
By analyzing the execution of the software using profilers, it is possible to identify some parts of decoding software requiring a high number of CPU clock ticks consisting in specific operations such as multiplications, divisions, calculation of the modulo, etc. Some of these operations can be cached in look-up tables by pre-computing all possible results and storing them in memory, so that later execution of these operations can be substituted by a mere access to memory.
The Intel C++ compiler [142] is used to compile the application and some optimization flags are used, (-O flags and SSE3 directives). Since writing the decoded frames to the Hard Disk (HD) is a process that slows down the execution, the frames are only saved at the end of the program so as to get a correct estimate of the processing time

Multithreading on Band Level

In this approach, the software is modified to place the decoding bulk of each band of a WZ frame in a different thread. The results are worse than those obtained when applying the multithreading on frame level, since the number of bands in each WZ frame varies with the consequence that the number of threads is sometimes smaller than the number of available core processors, which obviously leads to a poorer performance.

Multithreading on the Band Level and Multithreading for Motion Interpolation

Profiling revealed that some functions appearing in the motion interpolation part of the software are a bottleneck and can be parallelized. Since the solution involving multithreading on the band level is leaving some processors unused, multithreading for the motion interpolation part of the algorithm is added.

8.3.4 Optimization results

The speed gains for the optimization solutions are depicted in Figure 8.6 and 8.7. The video sequences used for testing are Soccer@15Hz, QCIF spatial resolution, and Soccer@30Hz, CIF spatial resolution. The simulations are performed on a multiprocessor computer with 4 Dual Core Intel Xeon processors and 8 GB of RAM.

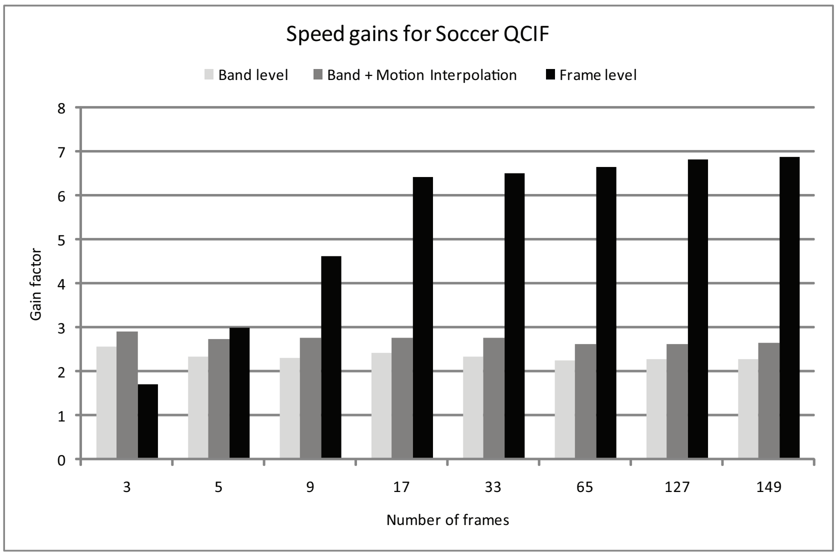


Figure 8.6: Decoding optimization gains for *Soccer*.

By analyzing the plots, the solution that provides the largest gains in terms of decoding time is multithreading on the frame level, which indeed runs up to 6 times faster than the initial DVC decoder. The speed-up gains of multithreading on the band level, and multithreading on motion interpolation are 2.5 times faster. Finally, the speed gain obtained by applying only multithreading on the band level is smaller, since it is 1.7 times faster than the initial codec.

8.4 Practical Implementation

This section addresses some practical aspects in the implementation of the DVC client and server demonstrator. The use of some software in the implementation

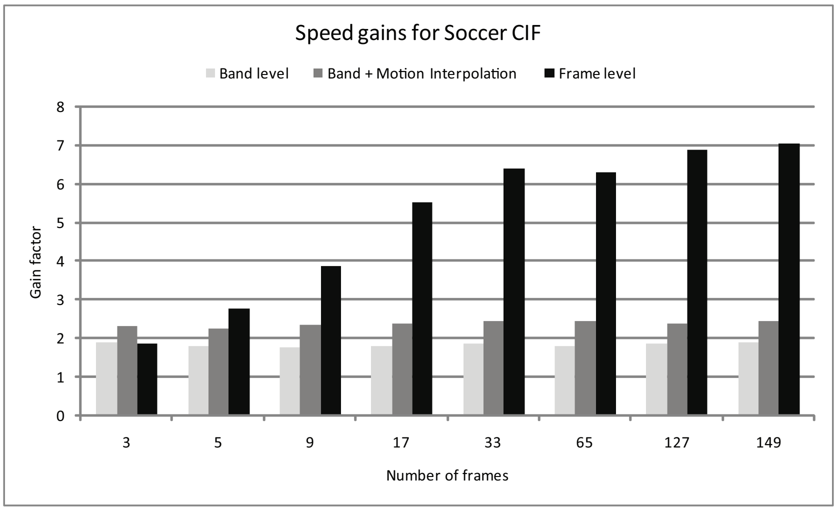


Figure 8.7: Decoding optimization gains for *Soccer*.

of the client and the server demonstrator is justified.

8.4.1 Software Options

Video Capturing Framework

The DVC client uses the Video For Windows (VFW) [143] framework developed by Microsoft to connect the capturing device to the client demonstrator application. There are other solutions, such as DirectShow [144], but they are not considered in the demonstrator since VFW provides a simple, complete and reliable way to connect external capturing devices and to play digital video in Microsoft Windows applications.

Graphical User Interface (GUI)

The graphical user interface of both client and server demonstrator is built using the Microsoft Foundation Classes library (MFC) [145]. This framework is a Microsoft library that wraps portions of the Windows API in C++ classes. There are several advantages for using this framework such as the creation of fast and small executables, the direct use of native Windows API's, fast compilation speed and big community support. On the other hand, the main disadvantage is the lack of portability to other operating systems, such as Linux and Macintosh. An alternative to MFC is the Fast Light ToolKit (FLTK) which is cross-platform operational and also provides a large set of functionalities.

JM and x264 Comparison

The initial DVC encoder [17] uses the JM [76] software to encode the key frames. This software does not achieve real time encoding. Therefore, it is replaced by the x264 [140] library which provides real time encoding capabilities. This is an open source library used in many popular applications. The high performance of x264 is attributed to the optimization of its rate control, motion estimation, macroblock mode decision, quantization and frame type decision algorithms. In addition, x264 uses assembly optimized code for many of the primitive operations. A comparison between the performance of the JM encoder and x264 shows that x264 is about 50 times faster and provides bit rates within 5% of JM for the same quality [146]. Figure 8.8 shows a comparison between the encoding frame rate of the JM encoder and x264 (both in Intra mode) for the video sequences *Soccer*, *Hallmonitor*, *Coastguard* and *Foreman*, in the QCIF resolution. The tests are performed on a Pentium 3.2 GHz Dual Core with 2 GB of RAM.

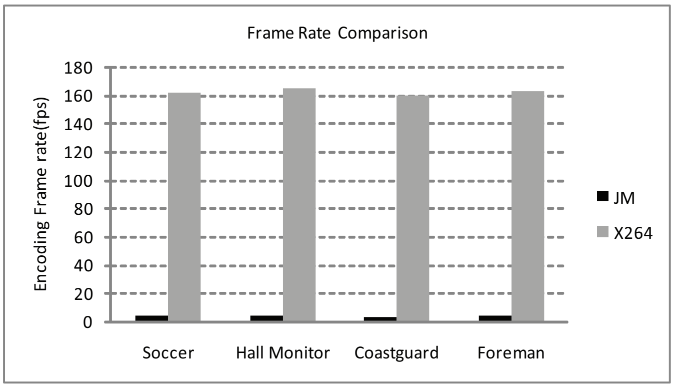


Figure 8.8: Encoding frame rate comparison of JM and x264 encoder, QP=34, QCIF resolution.

Figure 8.9 presents a comparison between the RD performance of the DVC codec using x264 and using JM encoders for the *Soccer*@15Hz and *Hallmonitor*@15Hz sequences. The results depicted in Figure 8.8 confirm what is stated in [146], the encoding frame rate of the x264 library is significantly higher than the one of JM accomplishing the requirements, i.e. realtime encoding. In Figure 8.9, the results show a decrease in the RD performance of the DVC codec when using x264. As reported in 8.8, there is an increase of about 5% in bit rate when comparing with JM. Nevertheless, this decrease in the RD performance is not considered critical for the demonstrator. With the x264 library realtime encoding is accomplished and with the JM library this cannot be achieved impeding the execution of a feasible demonstration.

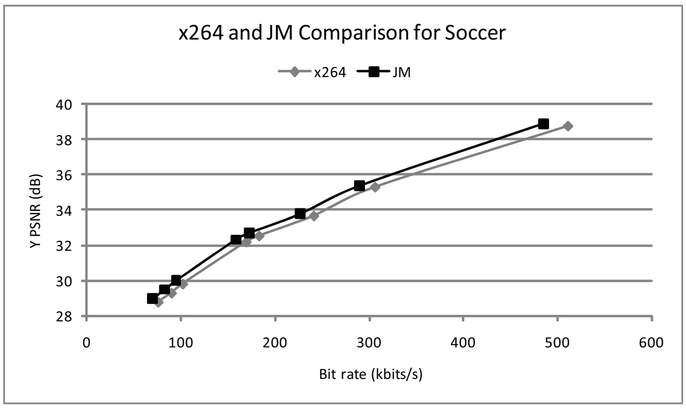
8.4.2 Description of the Graphical User Interface

The graphical conception for the client and server demonstrators is intended to be intuitive and easy to use. Both of these applications require an additional configuration file that controls some aspects of the DVC coding.

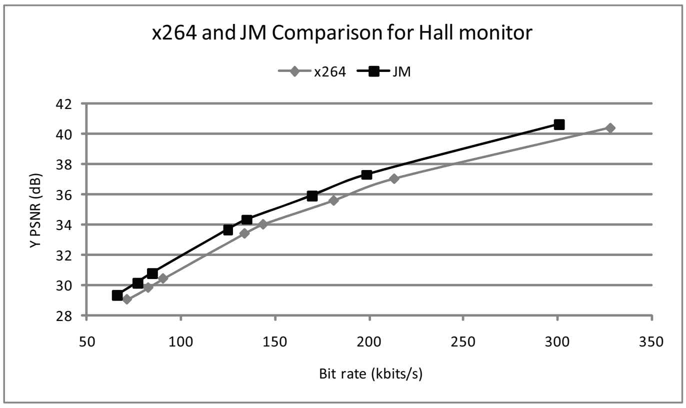
Client Demonstrator Interface

The main dialog box of the client demonstrator is depicted in Figure 8.10. Figures 8.11 illustrates the secondary dialog boxes related with the video device configuration.

In the main dialog box, the user defines the location of the encoder configuration

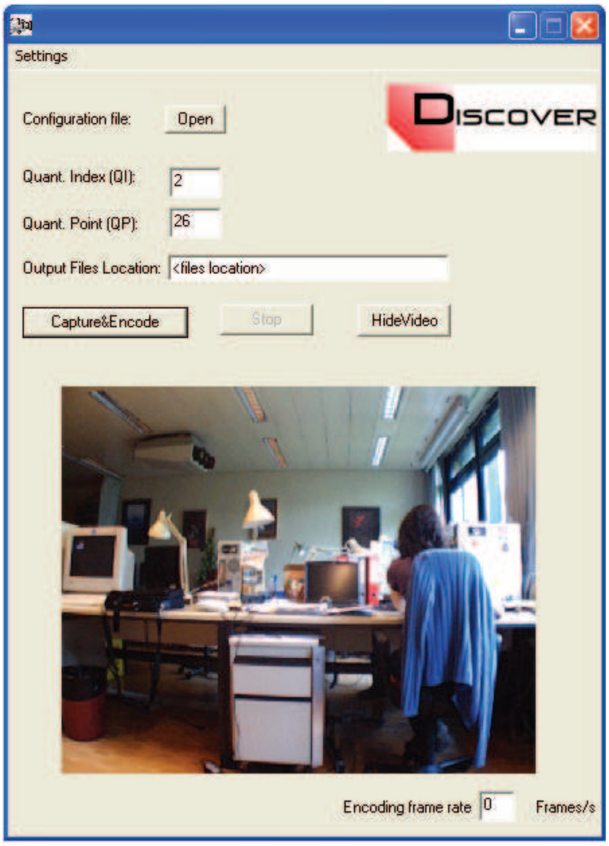


(a) *Soccer*



(b) *Hallmonitor*

Figure 8.9: RD performance comparison for the *Soccer* and *HallMonitor* sequences (QCIF, 15Hz).



Configuration file:

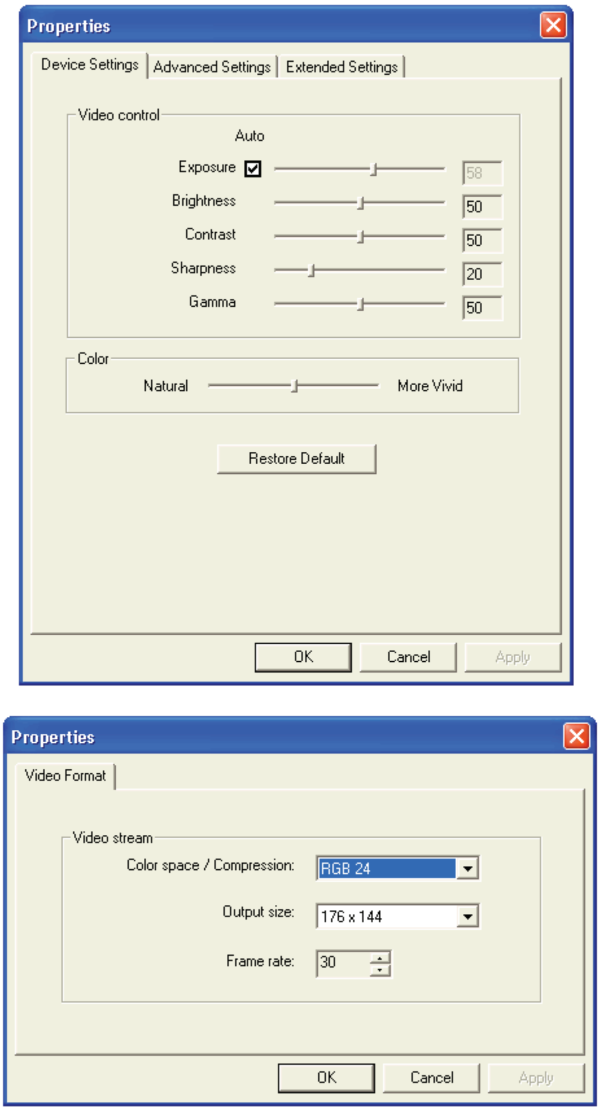
Quant. Index (QI): 2

Quant. Point (QP): 26

Output Files Location: <files location>

Encoding frame rate 0 Frames/s

Figure 8.10: Client demonstrator main dialog box.



Properties

Video control

Exposure Auto ☑ 58

Brightness 50

Contrast 50

Sharpness 20

Gamma 50

Color: Natural — More Vivid

Properties

Video stream

Color space / Compression: RGB 24

Output size: 176 x 144

Frame rate: 30

Figure 8.11: Video properties and format dialog boxes of the client demonstrator's client.

file. In addition, the user defines in the main dialog box, both the Quantization Index (QI) and the Quantization Point (QP) used for DVC encoding. Moreover, the output file location of the H.264/AVC Intra compressed bit stream and the generated parity bits should be defined as well.
The video properties dialog box is used to define the video spatial resolution, the frame rate and the color space of the captured frames. The video format dialog is used to define additional parameters such as the contrast, the brightness and gamma correction. Finally, as depicted in Figure 8.10, the video captured by the camera can be seen in real time in the lower part of the main dialog.

Server Demonstrator Interface

The server demonstrator interface is composed of a main dialog box and a pop-up window that allows the display of the decoded video sequence, both depicted in Figure 8.12.
In the main dialog box, the user must define the location of the decoder configuration file. In addition, the location of the AVC/H.264 Intra stream, the number of frames to decode and the number of threads used in the decoding process should be specified. It defines the number of simultaneous WZ frames to decode. The pop-up window serves to display, during the flow of the program, the decoded frames.

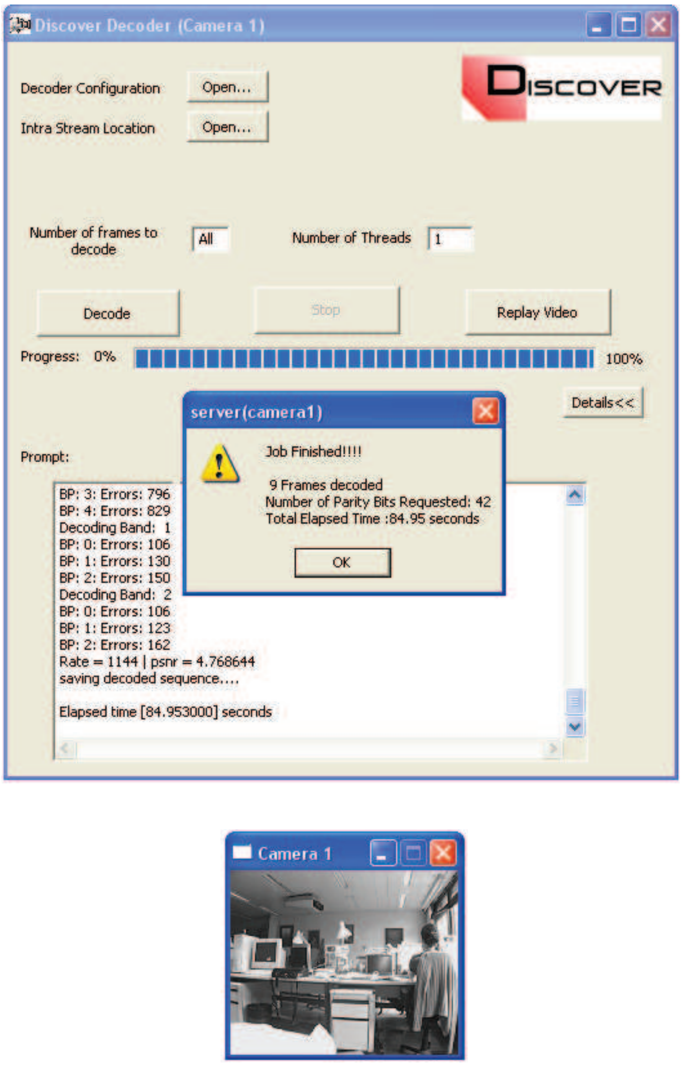
8.4.3 Decoding performance

Regarding the final implementation of the demonstrator, with the optimization techniques included in the software and using the x264 library, some simulations are made with the objective of evaluating the decoding time performance of the server demonstrator. The video sequences used are *Hallmonitor*, *Coastguard*, *Soccer* and *Foreman*, QCIF resolution, 15Hz and *Hallmonitor* CIF resolution, 30 Hz. The tests are realized using a multiprocessor computer with 4 Dual Core Intel Processors and 8 GB of RAM. In the server graphical application, the number of threads is set to 8, which means that the server demonstrator is decoding 8 frames simultaneously. Table 8.1 shows the obtained results.

Table 8.1: Server demonstrator decoding time for QI=4

Sequence	Resolution	Decoded frames	Decoding time (Seconds)	Average decoding frame rate (Frames/Second)
Hallmonitor	176x144	165	151.78	1.09
Coastgurad	176x144	150	197.41	0.76
Foreman	176x144	150	352.77	0.42
Soccer	176x144	150	491.81	0.30
Hallmonitor	352x288	165	690.08	0.24

The achieved results show that the decoding operation is still far from occurring in realtime, even after the inclusion of the optimization techniques. The results, obviously, show as well that for the same sequence, the time needed to decode



Number of frames to decode: All

Number of Threads: 1

Job Finished!!!!

9 Frames decoded
Number of Parity Bits Requested: 42
Total Elapsed Time :84.95 seconds

```
BP: 3: Errors: 796
BP: 4: Errors: 829
Decoding Band:  1
BP: 0: Errors: 106
BP: 1: Errors: 130
BP: 2: Errors: 150
Decoding Band:  2
BP: 0: Errors: 106
BP: 1: Errors: 123
BP: 2: Errors: 162
Rate = 1144 | psnr = 4.768644
saving decoded sequence....

Elapsed time [84.953000] seconds
```

Figure 8.12: Server demonstrator GUI

the sequence increases with the spatial resolution increase. These results confirm the well known WZ coding trade-off, where the encoding complexity benefits are paid in terms of decoding complexity as the optimized server demonstrator does not yet achieve realtime decoding.

8.5 Conclusion

This chapter presents the choices made to realize the DVC demonstrator. The main requirements for the demonstrator are presented as well as the limitations observed and the simplifications performed. Moreover, all the elements involved in the demonstration scenario are rationalized, including the hardware requirements, the network transmission and the software requirements. In this context, two main entities can be highlighted, the client and the server demonstrator. The client demonstrator is composed of four fundamental parts: video acquisition, DVC configuration, DVC encoding and data storage. The server demonstrator is also composed of four main parts: DVC configuration, data reading, DVC decoding and video display. The DVC codec used reveals some limitations that refer to the capability of encoding the key frames in real time and to the need for WZ realtime decoding. The first limitation is solved by replacing the AVC/H.264 Intra software used in the initial DVC software with one that provided realtime encoding. The second limitation given by the need for WZ realtime decoding, represents the main constraint of the software. Some optimization work has been done on the WZ video decoding part, with significant speedup gains. A presentation of the graphical user interfaces developed is given, and finally some experimental results are presented. Even though a multiprocessor computer and some optimization techniques are used, the WZ decoding is still far from operating in realtime for the moment.

The work demonstrates, even though the decoding is not real time, that it is actually possible to apply DVC in a complete coding service scenario removing all unrealistic assumptions such as original being available at the decoder. The demonstrator is fully configurable allowing the user to best tune the performance of the codec depending on the application scenario. Finally, this demonstration highlights the low encoding complexity of DVC but confirms the high decoding complexity of DVC.

Chapter 9

Conclusion

Different contributions related to Distributed Video Coding are presented in this thesis. Several approaches to improve the compression efficiency of DVC are presented. First, the improved SI is introduced for monoview DVC to generate a better prediction for an enhanced iterative reconstruction. More specifically, the improved SI exploits a partially decoded frame, which is obtained by DVC decoding with MCTI as SI, and the key frames to improve the quality of the SI, especially in areas with significantly large prediction error. Then, the re-estimated SI is used along with the decode WZ bins to enhance the reconstruction of the WZ frames. This includes suspicious vector detection, motion vector refinement and mode selection for the motion compensation phase. The results show that significant, up to 1.0 dB, gains are obtained for high motion video content. The improved SI tends to bring more improvement for parts of the video where MCTI fails to well-estimate the motion. Thus, the objective quality trough out the whole video is more regular with the improved SI. On the other hand, degradation of the decoded video quality in observed when only MCTI is used for video segments with high and irregular motion.

Similarly, the same concept is also applied to multiview DVC with a difference in the mode selection stage. In the latter, more reference frames can be used in the multiview scenario as frames from the side cameras are available at the decoder in addition to both previous and forward key frames. Moreover, suspicious motion vector detection is skipped for the multiview case as all the blocks are re-estimated. A maximum gain of around 0.9 dB is achieved over monoview DVC.

Then, fusion techniques between different predictions for multiview DVC are also proposed in this work. The first one is entirely performed at the decoder and uses the key frames as estimates of the WZ frame to make a pixel-basis decision on which SI candidate to use in the decoding process. The second fusion is encoder-driven as the encoder helps the decoder in performing the fusion by sending a binary mask to inform the decoder on which key frame pixel to use as a predictor. The fusions improve the performance over monoview DVC by a maximum of 1.0 dB, showing that a gain in compression efficiency is achieved by exploiting the inter-view correlation in addition the temporal one. Nevertheless, DVC's performance is still inferior to conventional coding in error-free environments. On the other hand, DVC has good error resilience properties, which is strengthened by some simulations results obtained in this research.

DVC actually exhibits better performance than conventional coding in error-prone environments. In the continuity, a hybrid error concealment for DVC is proposed, where the outcome of spatial EC is used as a partially decoded frame to improve the performance of temporal EC. The hybrid EC for DVC outperforms AVC/H.264 with EC in different coding modes.

Furthermore, DVC offers the tools to construct scalable codecs, which are codec-independent. This gives higher flexibility in the way the different layers are distributed in a network and helps in reducing the traffic at the same time in certain scenarios. The DVC-based scheme tend to decrease steadily in performance with error rate increase, where as conventional coding protected with RS codes tend to have a cliff effect (i.e. sudden significant drop in performance), when the limiting correcting capability of the RS code is reached.

Video surveillance is one of the most promising application scenarios for DVC. In this thesis, a transform domain scrambling scheme for DVC is prposed for such application scenarios and other privacy sensitive applications, where DVC could be used. The scrambling consists in either randomly inverting the signs of the DCT coefficients within regions of interest or randomly permuting the coefficients themselves. Moreover, the scrambling is shown to provide a good level of security without impairing the performance of the DVC scheme when compared to the one without scrambling.

Finally, the DVC demonstrator built during this research is described in details. It runs in realistic conditions as it does not require originals at the decoder but the decoding is still not real time for the moment even though a lot of effort was spent on optimizing the decoder's software.

The main drawbacks of DVC are its low coding efficiency and its highly complex decoding process. The latter is based on an iterative approach in addition to the expensive motion search task in terms of complexity. Moreover, the scheme requires a feedback mechanism for rate control, which introduces further delay. The results show that DVC is inferior in terms of compression efficiency to conventional coding even though it outperforms Intra coding in certain situations (i.e. low motion video content) but it is mainly outperformed by the predictive modes.

Therefore, future work should aim at tackling these problems so that we move further towards first feasible DVC solutions in terms of decoding complexity and delay. Furthermore, encoder rate control mechanisms would help in reducing the delay and the complexity as well since fewer decoding iterations are required. On the other hand, current solutions tend to highly overestimate the rate, which significantly reduces the overall performance. Finally, fast motion search techniques can be also used to reduce the decoder's complexity even further.

Acknowledgements

First, I would like to thank Dr. Shuiming Ye for his help with the work on the proposed improved side information and hybrid error concealment. Many thanks also to Mr. Carlos Bandeirinha for his help in the implementation of the demonstrator. Finally, I would like to thank Mr. Gary Sullivan and Dr. Alexis Tourapis for the helpful discussions regarding the parameters of the AVC Intra encoder. Finally, a special thanks to Dr. Frederic Dufaux for his help throughout the four years of my PhD thesis.

This work was partially supported by the European Project DISCOVER (IST Contract 015314) and the European Network of Excellence VISNET II (IST Contract 1-038398), both funded under the European Commission IST 6th Framework Program.

I would like also to acknowledge the use of the DISCOVER codec, a software which started from the IST WZ software developed at the Image Group from Instituto Superior Técnico (IST) of Lisbon by Catarina Brites, Joo Ascenso and Fernando Pereira.

Bibliography

[1] Moving Picture Experts Group (MPEG).
URL `http://www.chiariglione.org/mpeg/`

[2] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, J. Ostermann, Distributed Monoview and Multiview Video Coding: Basics, Problems and Recent Advances, IEEE Signal Processing Magazine, Special Issue on Signal Processing for Multiterminal Communication Systems (2007) 67–76.

[3] J. Slepian, J. Wolf, Noiseless Coding of Correlated Information Sources, IEEE Trans. on Information Theory vol 19 (4).

[4] A. Wyner, J. Ziv, The Rate-Distortion Function for Source Coding with Side Information at the Decoder, IEEE Transactions on Information Theory vol 22 (1).

[5] R. Puri, K. Ramchandran, PRISM: A New Robust Video Coding Architecture Based on Distributed Compression Principles, in: Allerton Conference on Communication, Control and Computing, 2002.

[6] B. Girod, A. Aaron, S. Rane, D. Rebollo-Monedero, Distributed Video Coding, in: Proceedings of the IEEE, Vol. vol 93, 2005, pp. 71–83.

[7] G. D. Forney, Coset Codes-Part I: Introduction and Geometrical Classification, IEEE Transactions on Information Theory 34 (1988) 1123–1151.

[8] G. D. Forney, Coset Codes-Part II: Binary Lattices and Related Codes, IEEE Transactions on Information Theory 34 (1988) 1152–1187.

[9] A. Aaron, R. Zhang, B. Girod, Wyner-Ziv Coding for Motion Video, in: Asilomar Conference on Signals, Systems and Computers, Pacific Grove, USA, 2002.

[10] C. Berrou, A. Glavieux, P. Thitimajshima, Near Shannon limit error-correcting coding and decoding: Turbo Codes, in: International Conference on Communications, Geneva, Switzerland, 1993.

[11] W. Pennebaker, J. Mitchell, JPEG: Still Image Compression Standard, Springer, Van Nostrand Reinhold, New York, 1992.

[12] A. Skodras, C. Christopoulos, T. Ebrahimi, The JPEG2000 Still Image Compression Standard, IEEE Signal Processing Magazine , Vol. 18 (5) (2001) 36–58.

[13] T. Wiegand, G. J. Sullivan, G. Bjntegaard, A. Luthra, Overview of the H.264/AVC Video Coding Standard, IEEE Trans. on Circuits and Systems for Video Technology , Vol. 13 (7) (2003) 560–576.

[14] JPEG Committee, JPEG XR Image Coding Specification, Tech. Rep. ISO/IEC 29199-2.

[15] S. Ye, Q. Sun, E. Chang, Edge Directed Filter based Error Concealment for Wavelet-based Images, in: IEEE (Ed.), International Conference on Image Processing (ICIP), Singapore, 2004.

[16] I. S. R. G. Solomon, Polynomial Codes Over Certain Finite Fields, Journal of the Society for Industrial and Applied Mathematics 8 (2) (1960) 300–304.

[17] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, M. Ouaret, The Discover Codec: Architecture, Techniques and Evaluation, in: Picture Coding Symposium (PCS), Lisboa, Portugal, 2007.

[18] B. Girod, A. Aaron, S. Rane, D. Rebollo-Monedero, Distributed Video Coding, Vol. 93, 2005, pp. 71–83.

[19] C.Tonoli, M.Dalai, P.Migliorati, R.Leonardi, Error Resilience Performance Evaluation of a Distributed Video Codec, in: Picture Coding Symposium (PCS), Lisbon, Portugal, 2007.

[20] J. Pedro, L. Soares, C. Brites, J. Ascenso, F. Pereira, C. Bandeirinha, S. Ye, F. Dufaux, T. Ebrahimi, Studying Error Resilience Performance for a Feedback Channel Based Transform Domain Wyner-Ziv Video Codec, in: Picture Coding Symposium (PCS), Lisbon, Protugal, 2007.

[21] M. Ouaret, F. Dufaux, T. Ebrahimi, Codec-Independent Scalable Distributed Video Coding, in: International Conference on Image Processing (ICIP), San Antonio, USA, 2007.

[22] G. Ungerboeck, Channel Coding with Multilevel/Phase Signals, IEEE Transactions on Information Theory 28 (1982) 55–67.

[23] M. Dalai, R. Leonardi, L. Torres, X. Artigas, F. Pereira, DISCOVER Deliverable 15: System Architecture Updates, Tech. rep.

[24] J. Ascenso, C. Brites, F. Pereira, Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding, in: EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, Slovak, 2005.

[25] K. Misra, S. Karande, H. Radha, Multi-hypothesis based Distributed Video Coding using LDPC codes, in: Allerton Conference on Commun, Control and Computing, 2005.

[26] A. Aaron, S. Rane, B. Girod, Wyner-Ziv Video Coding with Hash-Based Motion Compensation at the Receiver, in: IEEE (Ed.), International Conference on Image Processing (ICIP), Singapore, 2004.

[27] E. Martinian, A. Vetro, J. Ascenso, A. Khisti, D. Malioutov, Hybrid Distributed Video Coding using SCA Codes, in: IEEE International Workshop Multimedia Signal Processing, Victoria, Canada, 2006, pp. 258–261.

[28] M. Maitre, C. Guillemot, L. Morin, 3-D Model-based frame interpolation for Distributed Video Coding of static scenes, IEEE Transactions on Image Processing 16 (5) (2007) 1246–1257.

[29] C. Brites, J. Ascenso, F. Pereira, Studying Temporal Correlation Modeling for Pixel based Wyner-Ziv Video Coding, in: International Conference on Image Processing, 2006, pp. 273–276.

[30] H. Wang, N. Cheung, A. Ortega, A Framework for Adaptive Scalable Video Coding using Wyner-Ziv Techniques, EURASIP Journal on Applied Signal Processing (60971).

[31] D. Kubasov, K. Lajnef, C. Guillemot, A Hybrid Encoder/Decoder Rate Control for a Wyner-Ziv Video Codec with a Feedback Channel, in: IEEE Multimedia Signal Processing Workshop, Chania, Crete, Greece, 2007.

[32] M. Tagliasacchi, J. Pedro, F. Pereira, S. Tubaro, An Efficient Request Stopping Method At The Turbo Decoder In Distributed Video Coding, in: EURASIP European Signal Processing Conference, Poznan, Poland, 2007.

[33] M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites, F. Pereira, Intra Mode Decision based on Spatio-Temporal Cues in Pixel Domain Wyner-Ziv Video Coding, in: IEEE International Conference on Acoustics and Speech Signal Processing, Vol. 2, Toulouse, France, 2006, pp. 57–60.

[34] M. Tagliasacchi, L. Frigerio, S. Tubaro, Rate Distortion Analysis of Motion Compensated Interpolation at the Decoder in Distributed Video Coding, IEEE Signal Processing Letters 14 (9) (2007) 625–628.

[35] S. Klomp, Y. Vatis, C. Brites, X. Artigas, L. Torres, F. Dufaux, D. Kubasov, M. Dalai, DISCOVER Deliverable 5: Reference Video Codec Specification from the Literature, Tech. rep.

[36] C. Brites, J. Ascenso, F. Pereira, Improving Transform Domain Wyner-Ziv Video Coding Performance, in: International Conference on Acoustics, Speech and Signal Processing, Toulouse, France, 2006.

[37] H. Malvar, A. Hallapuro, M. Karczewicz, L. Kerofsky, Low-complexity Transform and Quantisation in H.264 / AVC, IEEE Transactions on Circuits and Systems for Video Technology 12 (7) (2003) 598– 603.

[38] W. W. Peterson, D. T. Brown, Cyclic Codes for Error Detection, in: Proceedings of the IRE, Vol. 49, 1961, pp. 228–235.

[39] A. Aaron, S. Rane, E. Setton, B. Girod, Transform-domain Wyner-Ziv codec for video, in: SPIE (Ed.), Visual Communications and Image Processing (VCIP), San Jose, California, USA, 2004.

[40] B. Vucetic, J. Yuan, Turbo Codes principles and applications, Kluwer Academic, USA, 2000.

[41] D. Rowitch, L. Milstein, On the Performance of Hybrid FEC/ARQ Systems using Rate Compatible Punctured Turbo Codes, IEEE Transactions on Communications 48 (6) (2000) 948–959.

[42] W. E. Ryan, A Turbo Code Tutorial, Tech. rep., New Mexico State University (1998).

[43] L. Bahl, J. Cocke, F. Jelinek, J. Raviv, Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate, IEEE Transactions on Information Theory 20 (1974) 284–287.

[44] P. Robertson, E. Villebrun, P. Hoeher, Comparison of Optimal and Suboptimal MAP Decoding Algorithms Operating in the Log Domain, in: International Conference on Communications (ICC), Washington, USA, 1995, pp. 1009–1013.

[45] T. Stockhammer, M. H. Hannuksela, H.264/AVC video for wireless transmission, IEEE Wireless Communications 12 (4) (2005) 6–13.

[46] S. Wenger, Proposed Error Patterns for Internet Experiments, Tech. Rep. Doc. VCEG Q15-I-16R1, VCEG Meeting, Red Bank, NJ, USA (October 1999).

[47] F. Pereira, L. Torres, C. Guillemot, T. Ebrahimi, R. Leonardi, S. Klomp, Distributed Video Coding: Selecting The Most Promising Application Scenarios, Signal Processing: Image Communication 23 (5) (2008) 339–352.

[48] F. Pereira, P. Correia, J. Ascenso, E. Acosta, L.Torres, C. Guillemot, C. Bandeirinha, M. Ouaret, F. Dufaux, T. Ebrahimi, R. Leonardi, M. Dalai, S. Klomp, DISCOVER Deliverable 19: Application Scenarios and Functionalities for DVC, Tech. rep.

[49] Surveillance Definition.
URL `http://en.wikipedia.org/wiki/Surveillance`

[50] Baby surveillance.
URL `http://www.homesecuritystore.com/babymonitor.html`

[51] Tiny Surveillance Camera.
URL `http://www.pimall.com/nais/e.menu-b.html`

[52] Wild Life Monitoring.
URL `http://www.ecpe.vt.edu/news/ar03/video.html`

[53] F. Hagemller, M. P. Manns, H. G. Musmann, J. F. Riemann, Medical Imaging in Gastroenterology and Hepatology, in: Series Falk Symposium, Vol. 124, Kluwer Academic, 2002.

[54] S. Srinivasan, C. Tu, Z. Zhou, D. Ray, S. Regunathan, G. Sullivan, An Introduction to the HD Photo Technical Design, Tech. Rep. JPEG document wg1n4183, Microsoft Corporation (April 2007).

[55] JPEG Committee.
URL http://www.jpeg.org/committee.html

[56] Joint Video Team (JVT).
URL http://www.itu.int/ITU-T/studygroups/com16/jvt/

[57] D. Marpe, V. George, H. L. Cycon, Performance Evaluation of Motion-JPEG2000 in Comparison with H.264/AVC Operated in Intra Coding Mode, in: Wavelet Applications in Industrial Processing, SPIE, Rhode Island, USA, 2003.

[58] D. Marpe, S. Gordon, T. Wiegand, H.264/MPEG4-AVC Fidelity Range Extensions: Tools, Profiles, Performance, and Application Areas, in: International Conference on Image Processing (ICIP), IEEE, Genova, Italy, 2005.

[59] P. Topiwala, Comparative Study of JPEG2000 and H.264/AVC FRExt I-Frame Coding on High-Definition Video Sequences, in: Optical Information Systems III, Vol. 5909, SPIE, 2005, pp. 284–292.

[60] Joint Video Team of ITU-T and ISO/IEC, Performance Comparison of Intra-only H.264/AVC HP and JPEG2000 for a Set of Monochrome ISO/IEC Test Images, Tech. rep., Document JVT-M014 (October 2004).

[61] M. Ouaret, F. Dufaux, T. Ebrahimi, On comparing JPEG2000 and Intraframe AVC, in: SPIE (Ed.), Applications of Digital Image Processing XXIX, San Diego, USA, 2006.

[62] M. Ouaret, F. Dufaux, T. Ebrahimi, On comparing image and video compression algorithms, in: International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM), Scottsdale, Arizona, USA, 2007.

[63] F. D. Simone, M. Ouaret, F. Dufaux, A. G. Tescher, T. Ebrahimi, A comparative study of JPEG 2000, AVC/H.264, and HD Photo, in: SPIE (Ed.), Applications of Digital Image Processing XXX, San Diego, CA, USA, 2007.

[64] K. R. Rao, P. Yip, Discrete Cosine Transform: Algorithms, Advantages, Applications, Academic Press Professional, 1990.

[65] D. A. Huffman, A Method for the Construction of Minimum Redundancy Codes, in: Institute of Radio Engineers, Vol. 40, 1962, pp. 1098–1101.

[66] W. B. Pennebaker, J. L. Mitchell, Arithmetic Coding Articles, IBM J. Res. Dev 32 (6) (1988) 717–774.

[67] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, Image Coding Using Wavelet Transform, IEEE Transactions on Image Processing 1–2 (1992) 205–219.

[68] A. Said, Introduction to Arithmetic Coding - Theory and Practice, Tech. Rep. HPL-2004-76, Imaging Systems Laboratory, HP Laboratories, Palo Alto, USA (April 2004).

[69] G.Bjontegaard, K. Lillevold, Context-Adaptive VLC Coding of Coefficients, Tech. Rep. JVT-C028 (May 2002).

[70] D. Marpe, G. Blattermann, T. Wiegand, Adaptive Codes for H.26L, Tech. rep. (January 2001).

[71] Joint Video Team of ITU-T and ISO/IEC, Draft Text of H.264/AVC Fidelity Range Extensions Amendment, Tech. rep., Document JVT-L047 (September 2004).

[72] H. S. Malvar, D. H. Staelin, The LOT: Transform Coding Without Blocking Effects, IEEE Transactions on Acoustics, Speech, and Signal Processing , Vol. 37 (4) (1989) 553–559.

[73] T. D. Tran, J. Liang, C. Tu, Lapped Tansform via Time-Domain Pre- and Post-Filtering, IEEE Transactions on Signal Processing , Vol. 51 (6) (2003) 1557–1571.

[74] JPEG software, Independent JPEG Group.
URL `http://www.ijg.org`

[75] JPEG2000 Kakadu Software.
URL `http://www.kakadusoftware.com/`

[76] AVC/H.264 software.
URL `http://iphome.hhi.de/suehring/tml/`

[77] Microsoft, HD Photo: Device Porting Kit Specification version 1.0 (June 2006).

[78] Y. Wang, Q. Zhu, Error Control and Concealment for Video Communication: A Review, in: Proceedings of the IEEE, Vol. 86, 1989, pp. 974–997.

[79] O. Hadar, M. Huber, R. Huber, S. Greenberg, New Hybrid Error Concealment for Digital Compressed Video, EURASIP Journal on Applied Signal Processing (12) (2005) 1821–1833.

[80] L. Wei, Y. Zhao, A. Wang, Improved Side-Information in Distributed Video Coding, in: International Conference on Innovative Computing, Information and Control, Beijing, China, 2006.

[81] M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites, F. Pereira, Exploiting Spatial Redundancy in Pixel Domain Wyner-Ziv Video Coding, in: IEEE (Ed.), International Conference on Image Processing (ICIP), Atalanta, USA, 2006.

[82] J. Ascenso, C. Brites, F. Pereira, Motion Compensated Refinement for Low Complexity Pixel Based Distributed Video Coding, in: IEEE (Ed.), International Conference on Advanced Video and Signal Based Surveillance, Sardinia, Italy, 2005.

[83] X. Artigas, L. Torres, Iterative Generation of Motion-compensated Side Information for Distributed Video Coding, in: IEEE (Ed.), International Conference on Image Processing (ICIP), Genova, Italy, 2005.

[84] W. A. R. J. Weerakkody, W. A. C. Fernando, J. L. Martnez, P. Cuenca, F. Quiles, An Iterative Refinement Technique for Side Information Generation in DVC, in: IEEE (Ed.), International Conference on Multimedia and Expo, 2007.

[85] W. Zhu, Y. Wang, Q. Zhu, Second-order Derivative based Smoothness Measure for Error Concealment in DCT based Codecs, IEEE Transactions on Circuits and Systems for Video Technology 8 (6) (1998) 713–718.

[86] W. Zeng, B. Liu, Geometric-structure-based Error Concealment with Novel Applications in Block-based Low-Bit-Rate Coding, IEEE Transactions on Circuits and Systems for Video Technology 9 (4) (1999) 648–665.

[87] X. Li, M. Orchard, Novel Sequential Error-concealment Techniques using Orientation Adaptive Interpolation, IEEE Transactions on Circuits and Systems for Video Technology 12 (10) (2002) 857– 864.

[88] M. Chen, Y. F. Zheng, M. Wu, Classification-based Spatial Error Concealment for Visual Communications, EURASIP Journal on Applied Signal Processing, Special Issue on Video Analysis and Coding for Robust Transmission, 2006.

[89] S. Ye, X. Lin, Q. Sun, Content based Error Detection and Concealment for Image Transmission over Wireless Channel, in: IEEE (Ed.), International Symposium on Circuits and Systems, Bangkok, Thailand, 2003.

[90] P. J. Lee, H. H. Chen, L. G. Chen, A New Error Concealment Algorithm for H.264 Video Transmission, in: IEEE (Ed.), International Symposium on Intelligent Multimedia, Video, and Speech Processing, Hong Kong, 2004.

[91] Y. Chen, K. Yu, J. Li, S. Li, An Error Concealment Algorithm for Entire Frame Loss in Video Transmission, in: Picture Coding Symposium (PCS), San Francisco, USA, 2004.

[92] Y. Chen, O. C. Au, C. W. Ho, J. Zhou, Spatio-Temporal Boundary Matching Algorithm for Temporal Error Concealment, in: IEEE (Ed.), International Symposium on Circuits and Systems, Greece, 2006.

[93] D. Agrafiotis, D. R. Bull, T. Chiew, P. Ferre, A. Nix, Enhanced Error Concealment for Video Transmission over WLANS, in: International Workshop on Image Analysis for Multimedia Interactive Services, Switzerland, 2005.

[94] C. Yim, W. Kim, H. Lim, Hybrid Error Concealment Method for H.264 Video Transmission over Wireless Networks, in: International Conference on Wireless Communications and Mobile Computing, Hawaii, USA, 2007.

[95] R. Bernardini, M. Fumagalli, M. Naccari, R. Rinaldo, M. Tagliasacchi, S. Tubaro, P. Zontone, Error Concealment Using a DVC Approach for Video Streaming Applications, in: EURASIP (Ed.), European Signal Processing Conference, Poznan, Poland, 2007.

[96] J. Zhai, K. Yu, J. Li, S. Li, A Low Complexity Motion Compensated Frame Interpolation Method, in: EEE (Ed.), International Symposium on Circuits and Systems, Kobe, Japan, 2005.

[97] K. P. Lim, G. Sullivan, T. Wiegand, Text Description of Joint Model Reference Encoding Methods and Decoding Concealment Methods, Tech. Rep. JVT-K049, Munich, Germany (March 2004).

[98] Free Viewpoint Television (FTV).
URL `http://www.tanimoto.nuee.nagoya-u.ac.jp/study/FTV/`

[99] X. Artigas, E. Angeli, L. Torres, Side Information Generation for Multiview Distributed Video Coding Using a Fusion Approach, in: 7th Nordic Signal Processing Symposium (NORSIG), Reykjavik, Iceland, 2006.

[100] X. Guo, Y. Lu, F. Wu, W. Gao, S. Li, Distributed Multiview Video Coding, in: SPIE (Ed.), Visual Communications and Image Processing (VCIP), San Jose, USA, 2006.

[101] X. Guo, Y. Lu, F. Wu, D. Zhao, W. Gao, Wyner Ziv based Multiview Video Coding, IEEE Transactions on Circuits and Systems for Video Technology 18 (6) (2008) 713–724.

[102] M. Flierl, B. Girod, Coding of Multiview Image Sequences with Video Sensors, in: IEEE (Ed.), International Conference on Image Processing (ICIP), Atlanta, GA, USA, 2006.

[103] M. Flierl, B. Girod, Video Coding with Motion-Compensated Lifted Wavelet Transforms, EURASIP Journal on Image Communication, Special Issue on Subband/Wavelet Interframe Video Coding 19 (7) (2004) 561–575.

[104] F. Dufaux, M. Ouaret, T. Ebrahimi, Recent Advances in Multiview Distributed Video Coding, in: SPIE Defense and Security Symposium, Mobile Multimedia/Image Processing for Military and Security Applications, Orlando, USA, 2007.

[105] M. Ouaret, F. Dufaux, T. Ebrahimi, Multiview Distributed Video Coding with Encoder Driven Fusion, in: European Conference on Signal Processing (EUSIPCO), Poznan, Poland, 2007.

[106] F. Dufaux, J. Konrad, Efficient, Robust, and Fast Global Motion Estimation for Video Coding, IEEE Transactions on Image Processing , Vol. 9 (3) (2000) 497–501.

[107] E. Martinian, A. Behrens, J. Xin, A. Vetro, View Synthesis for Multiview Video Compression, in: Picture Coding Symposium (PCS), 2006.

[108] S. M. Seitz, C. R. Dyer, View Morphing, in: SIGGRAPH, 1996, pp. 21–30.

[109] X. Artigas, F. Tarres, L. Torres, Comparison of Different Side Information Generation Methods for Multiview Distributed Video Coding, in: International Conference on Signal Processing and Multimedia Applications (SIGMAP), Barcelona, Spain, 2007.

[110] S. Ye, M. Ouaret, F. Dufaux, T. Ebrahimi, Improved Side Information Generation with Iterative Decoding and Frame Interpolation for Distributed Video Coding, in: IEEE (Ed.), International Conference on Image Processing (ICIP), San Deigo, USA, 2008.

[111] Joint Bi-level Image experts Group.
URL `http://www.jpeg.org/jbig/`

[112] R. M. Haralick, S. R. Stemberg, X. Zhuang, Image Analysis using Mathematical Morphology, IEEE Transactions on Pattern Analyis and Machine Intelligence , Vol 9 (4) (1987) 523–550.

[113] F. Dufaux, Multigrid Block Matching Motion Estimation for Generic Video Coding, Ph.D. thesis, Ecole Polytechnique Federale de Lausanne (1994).

[114] M. Z. Coban, R. M. Mersereau, Fast Rate-Constrained N-step Search Algorithm for Motion Estimation, in: IEEE (Ed.), International Conference on Acoustics, Speech and Signal Processing, Seattle, WA, USA, 1998.

[115] Introduction to SVC Extension of Advanced Video Coding, Tech. Rep. ISO/IEC JTC1/SC29/WG11, International Organization for Standardization, Coding of Moving Pictures and Audio, Poznan, Poland (July 2005).

[116] Scalable Video Coding.
URL `http://www.chiariglione.org/mpeg/technologies/mp04-svc`

[117] J. R. Ohm, Complexity and delay analysis of MCTF interframe wavelet structures, Tech. Rep. ISO/IEC JTC1/SC29/WG11, Document M8520 (July 2002).

[118] M. Flierl, Video Coding with Lifted Wavelet Transforms and Frame-Adaptive Motion Compensation, in: VLBV, 2003.

[119] M. Tagliasacchi, A. Majumdar, K. Ramchandran, A Distributed Source Coding based Spatio-Temporal Scalable Video Codec, in: Picture Coding Symposium (PCS), San Francisco, USA, 2004.

[120] A. Sehgal, A. Jagmohan, N. Ahuja, Scalable Video Coding Using Wyner-Ziv Codes, in: Picture Coding Symposium (PCS), San Francisco, USA, 2004.

[121] H. Wang, N.-M. Cheung, A. Ortega, A Framework for Adaptive Scalable Video Coding Using Wyner-Ziv Techniques, EURASIP Journal on Applied Signal Processing 2006 (2008) 267.

[122] T. Ebrahimi, F. Pereira, The MPEG-4 Book, Prentice Hall, 2002.

[123] F. Dufaux, G. Baruffa, F. Frescura, D. Nicholson., JPWL - an Extension of JPEG 2000 for Wireless Imaging, in: International Symposium on Circuits and Systems, Island of Kos, Greece, 2006.

[124] I. Moccagatta, S. Soudagar, J. Liang, H. Chen, Error-Resilient Coding in JPEG-2000 and MPEG-4, IEEE Journal on Selected Areas in Communications 18 (6) (2000) 899–914.

[125] D. Taubman, M. Marcellin, JPEG 2000: Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publishers, 2002.

[126] D. Santa-Cruz, R. Grosbois, T. Ebrahimi, JPEG2000 Performance Evaluation and Assessment, Signal Processing: Image Communication 17 (1) (2002) 113–130.

[127] A. Bilgin, Z. Wu, M. Marcellin, Decompression of Corrupt JPEG2000 Codestreams, in: Data Compression Conference, Snowbird, Utah, USA, 2003.

[128] Joint Scalable Verification Model (JSVM).
URL `http://ip.hhi.de/imagecomG1/savce/downloads`

[129] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Y. Tian, A. Ekin, Blinkering Surveillance: Enabling Video Privacy through Computer Vision, Tech. Rep. RC22886, IBM (2003).

[130] E. Newton, L. Sweeney, B. Malin, Preserving Privacy by De-identifying Facial Images, Tech. Rep. CMU-CS-03-119, Carnegie Mellon University (2003).

[131] D. A. Fidaleo, H. A. Nguyen, M. Trivedi, The Networked Sensor Tapestry (NeST): A Privacy Enhanced Software Architecture for Interactive Analysis of Data in Video-Sensor Networks, in: Proceedings of the ACM 2nd Int. Workshop on Video Surveillance and Sensor Networks, New York, 2004.

[132] F. Dufaux, T. Ebrahimi, Toward a secure JPEG, in: Applications of Digital Image Processing XXIX, Vol. vol 6312, 2006.

[133] F. Dufaux, M. Ouaret, Y. Abdeljaoued, A. Navarro, F. Vergnenegre, T. Ebrahimi, Privacy Enabling Technology for Video Surveillance, in: Mobile Multimedia/Image Processing for Military and Security Applications, Vol. vol 6250, 2006.

[134] F. Dufaux, T. Ebrahimi, Scrambling for Video Surveillance with Privacy, in: IEEE Proceedings of the workshop on Privacy Research in Vision, IEEE Computer Society, 2006.

[135] F. Dufaux, T. Ebrahimi, Region-Based Transform-Domain Video Scrambling, in: Visual Communications and Image Processing (VCIP), Vol. vol 6077, 2006.

[136] T. Boult, PICO: Privacy through Invertible Cryptographic Obscuration, in: IEEE/NFS (Ed.), Workshop on Computer Vision for Interactive and Intelligent Environments, 2005.

[137] F. Dufaux, T. Ebrahimi, Recent Advances in MPEG-7 Cameras, in: SPIE (Ed.), Proceedings Applications of Digital Image Processing XXIX, San Diego, CA, USA, 2006.

[138] F. Dufaux, T. Ebrahimi, H.264/AVC Video Scrambling for Privacy Protection, in: IEEE International Conference on Image Processing, San Diego, USA, 2008.

[139] D. Knuth, The Art of Computer Programming, 2nd Edition, Addison-Wesley, 1981.

[140] x264 Software.
URL http://www.videolan.org/developers/x264.htm

[141] NetBios.
URL http://en.wikipedia.org/wiki/NetBIOS

[142] Intel-based compiler.
URL http://www.intel.com/cd/software/products/asmo-na/eng/compilers/279578.htm

[143] Video For Windows (VFW).
URL http://en.wikipedia.org/wiki/VideoforWindows

[144] Direct Show.
URL http://en.wikipedia.org/wiki/DirectShow

[145] MFC.
URL http://msdn2.microsoft.com/en-us/library/d06h2x6e(vs.71).aspx

[146] L. Merritt, X264: A high performance H.264/AVC encoder, Tech. rep.